

Parallel Robots: Theory and Applications

Patrick Grosch  
Federico Thomas

---

# Parallel Robots With Unconventional Joints

Kinematics and Motion Planning



# **Parallel Robots: Theory and Applications**

## **Series Editors**

J. P. Merlet, Sophia Antipolis, France

Sébastien Briot, Institut de Recherche en Communications et Cybernétique de Nantes, CNRS, Nantes, France

P. Martinet, Nantes, France

Since a few decades, parallel robots have attracted the attention of industrials and researchers. For industrials, they are interesting due to their inherent advantages which are their large payload-to-mass ratio, intrinsic stiffness properties and great acceleration capacities. For researchers, their inherent complexity leading to issues in terms of design, modelling and control allowed the definition of many interesting scientific problems to be solved. We reached now a point where the research on parallel robots has led to the definition of a quantity of scientific results which, for the moment, miss to be compiled to be easily usable by researchers, teachers, professionals and students for their formation and future works. This collection aims at filling this gap, by proposing a series of book devoted to all the scientific and technological fields required for having a better understanding of the behavior of parallel robots and/or for helping the future parallel robot designer in his work. Therefore, we intend to cover all topics on parallel robots, from their (geometric, kinematic, dynamic, elastic, etc.) modelling, to their advanced control, by also considering the singularity analysis, the calibration and model parameter identification problems, the definition of advanced algebraic tools necessary for their study, their optimal design, and many other scientific and technological aspects. Springer and the series editors welcome book proposals. Potential authors are invited to contact Nathalie Jacobs, Executive Publishing Editor Engineering, at [nathalie.jacobs@springer.com](mailto:nathalie.jacobs@springer.com)

More information about this series at <http://www.springer.com/series/13855>

Patrick Grosch · Federico Thomas

# Parallel Robots With Unconventional Joints

Kinematics and Motion Planning



Springer

Patrick Grosch  
Institut de Robòtica i Informàtica  
Industrial, CSIC-UPC  
Barcelona, Spain

Federico Thomas  
Institut de Robòtica i Informàtica  
Industrial, CSIC-UPC  
Barcelona, Spain

ISSN 2524-6232

ISSN 2524-6240 (electronic)

Parallel Robots: Theory and Applications

ISBN 978-3-030-11303-2

ISBN 978-3-030-11304-9 (eBook)

<https://doi.org/10.1007/978-3-030-11304-9>

Library of Congress Control Number: 2019932702

© Springer Nature Switzerland AG 2019

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG  
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

# Preface

In this book, it is shown how the substitution of ordinary joints with unconventional joints, such as lockable revolute or non-holonomic spherical joints, in lower mobility parallel robots provide them with fascinating features. The most important of them being the possibility of approximating trajectories for their moving platform in a configuration space of higher dimension than the number of their continuously actuated joints. More specifically, it is shown how the introduction of these unconventional joints allows a parallel robot with an instantaneous kinematics equivalent to that of a lower mobility robot to attain any configuration within a six-dimensional workspace, that is, to behave similarly to a full-mobility robot at the cost of introducing maneuvers.

After presenting a general kinematic framework for lockable revolute joints ( $R_b$  joints, for short) and non-holonomic spherical joints ( $S_n$  joints, for short), three particular parallel robots are analyzed in detail; namely

- the  $4R_bRPS$  spatial reconfigurable robot (Chap. 2),
- the  $3S_nPU$  spatial non-holonomic robot (Chaps. 4 and 5), and
- the  $S-2SPS$  spherical non-holonomic robot (Chaps. 6 and 7).

The study of each of these three robots has been conducted with the greatest possible generality. In general, the performed analyses include

- the computation of its direct and inverse kinematics (both in position and velocity),
- the analysis of its singularities,
- the computation of its workspace,
- the design and implementation of a motion planning algorithm, and
- the implementation of a prototype to validate the theoretical results.

The methods used to perform the above analyses and computations, and the algorithms proposed to solve the path planning problems they originate, cover a wide range of situations that can be used as a reference to other parallel robots with the aforementioned unconventional joints.

The presented path planners are open-loop methods based on first-order kinematic models (the dynamics of the system is not considered). This obviously simplifies the problem to the point in which most of the presented mathematical models are derived from elementary geometrical and kinematical arguments. Nevertheless, these models are enough for many practical applications where inertial and gravity forces can be neglected compared to the forces exerted by the actuators, and provided that we are able to construct the input as a function of the system state to compensate for noises and errors in the system.

The methods and results presented in this book should be of interest to practicing and research engineers as well as Ph.D. students from the field of mechanical engineering and, in particular, from robotics.

All the material presented here is the result of a research started in 2008, when Patrick Grosch joined the *Institut de Robòtica and Informàtica Industrial* (CSIC-UPC) as development engineer. Due to financial shortages that drove Patrick to work in many different development projects, this research spanned over more than 8 years. After a lot of setbacks that challenged our capabilities and pushed our patience to the limit, this research eventually led Patrick to achieve the Ph.D. in 2016. The encountered difficulties were alleviated thanks to the kind help of our colleagues. In particular, we want to mention Raffaele Di Gregorio, from the University of Ferrara, Italy, who invited us to spend a fruitful research stay in his laboratory. We also want to extend our gratitude to Krzysztof Tchoń and Janusz Jakubiak, from the University of Wrocław, Poland. Without their help some parts of this book would not have been possible.

Barcelona, Spain  
May 2018

Federico Thomas

# Contents

<b>1</b>	<b>Introduction: Lockable and Non-holonomic Joints . . . . .</b>	1
1.1	Motivation . . . . .	1
1.1.1	Reconfigurable Robots with Lockable Joints . . . . .	6
1.1.2	Underactuated Robots with Non-holonomic Joints . . . . .	6
1.2	Precursors . . . . .	8
1.2.1	Precursors of Robots with Lockable Joints . . . . .	9
1.2.2	Precursors of Robots with Non-holonomic Joints . . . . .	10
1.3	Organization of this Book . . . . .	15
References	. . . . .	17
<b>2</b>	<b>Parallel Robots with Lockable Revolute Joints . . . . .</b>	19
2.1	Kinematics of the $4R_b$ RPS Parallel Robot . . . . .	21
2.1.1	Position Analysis . . . . .	22
2.1.2	Singularities . . . . .	23
2.2	Maneuvers . . . . .	25
2.3	Motion Planning . . . . .	27
2.3.1	Generating the Road Map . . . . .	28
2.3.2	Finding a Path . . . . .	31
2.4	Hardware Implementation . . . . .	31
2.5	Software Implementation . . . . .	33
References	. . . . .	34
<b>3</b>	<b>Spherical Non-holonomic Joints . . . . .</b>	35
3.1	Under-Actuated Parallel Robots with Spherical Non-holonomic Joints . . . . .	36
3.2	Implementation of Spherical Non-holonomic Joints . . . . .	38
References	. . . . .	42
<b>4</b>	<b>Kinematics of the <math>3S_n</math>PU Spatial Robot . . . . .</b>	43
4.1	The $3S_n$ PU Robot . . . . .	43
4.2	Instantaneous Kinematics . . . . .	45

4.3	Statics Analysis . . . . .	48
4.4	Singularities . . . . .	51
4.5	Controllability . . . . .	53
4.6	Example. . . . .	55
	References . . . . .	57
<b>5</b>	<b>Motion Planning for the <math>3S_nPU</math> Robot . . . . .</b>	<b>59</b>
5.1	Motion Planning . . . . .	59
5.2	Using Truncated Series . . . . .	63
5.3	Example. . . . .	64
	References . . . . .	66
<b>6</b>	<b>Kinematics of the <math>S_n</math>-2UPS Spherical Robot . . . . .</b>	<b>67</b>
6.1	Kinematic Model . . . . .	69
6.2	Deriving a Bilinear Model . . . . .	71
6.3	Singularities . . . . .	74
6.4	A, B, and Rotations in $\mathbb{R}^4$ . . . . .	75
6.5	Workspace Computation . . . . .	79
6.5.1	Graphical Representation of the Platform Orientation . . . . .	79
6.5.2	Workspace Boundaries Due to Singularities . . . . .	80
6.5.3	Workspace Boundaries Due to Joint Limits . . . . .	82
	References . . . . .	85
<b>7</b>	<b>Motion Planning for the <math>S_n</math>-2UPS Robot . . . . .</b>	<b>87</b>
7.1	Kinematic Model . . . . .	88
7.2	Three-Move Motion Planner . . . . .	89
7.3	Two-Move Motion Planner . . . . .	90
7.4	Single-Move Motion Planner . . . . .	92
7.5	Example. . . . .	95
7.5.1	Three-Move Motion Planner. . . . .	96
7.5.2	Two-Move Motion Planner . . . . .	96
7.5.3	Single-Move Motion Planner . . . . .	97
7.5.4	Comparing the Three Motion Planners . . . . .	99
7.6	Implementation. . . . .	101
	References . . . . .	102
<b>8</b>	<b>Conclusions . . . . .</b>	<b>105</b>

# Chapter 1

## Introduction: Lockable and Non-holonomic Joints

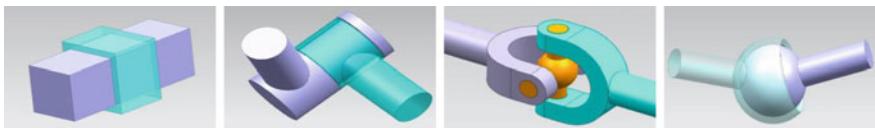


### 1.1 Motivation

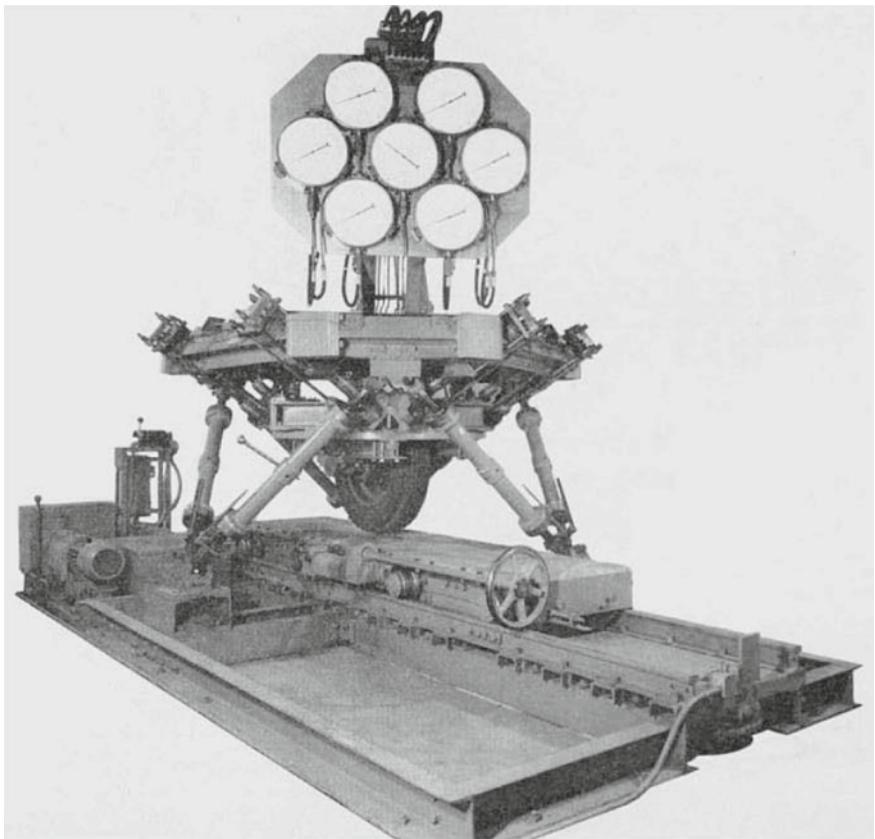
A serial robot is a set of rigid bodies, or links, connected in series through actuated joints, which are typically either revolute (R) or prismatic (P) joints. One end of this serial chain of links is called the base and the other the end-effector. In a parallel robot, the end-effector (also known in this case as the moving platform) is connected to the fixed base through several serial chains. In this case, most of the joints are not actuated. These passive joints are typically either universal (U) or spherical (S) joints. Prismatic, revolute, universal, and spherical joints constitute the four major joints used in robotics (Fig. 1.1).

The origin of parallel robots is attributed to Gough and Whitehall [16] (Fig. 1.2) and Stewart [37]. Their work led to what is nowadays known as the Gough–Stewart platform. In 1965, Stewart formalized the concepts that define a parallel robot. Since then, many developments and studies have been carried out on this kind of robots. Innocenti and Parenti-Castelli [21] and Dasgupta and Mruthyunjaya [11] give an introductory overview of the different kinds of parallel robots. The references compiled in [9, 27] permit to establish a good starting point where to find methods for the computation of the direct and inverse kinematics, the configuration space and the singularities, the manipulability and accuracy, etc., of different parallel robots.

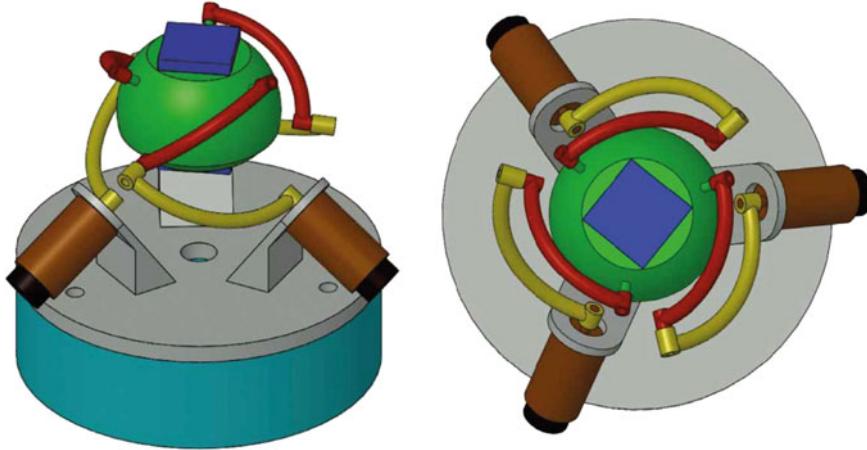
The Gough–Stewart platform remains as the most popular spatial parallel robot. It is commonly seen in most motion simulators. This robot is said to have a 6SPS architecture, meaning that each of its 6 legs consists of a prismatic joint connected to the fixed base and to the moving platform through passive spherical joints. The underline is used to denote that the joint is actuated. This parallel robot is said to be a *fully-parallel* robot because: (a) it has as many serial chains (known as legs) as the number of degrees of freedom of its moving platform, (b) each leg possesses only one actuated joint, and (c) no link of the legs is linked to more than two bodies [28, 34].



**Fig. 1.1** Major joints used in robotics. From left to right: Prismatic (P), revolute (R), universal (U), and spherical (S)



**Fig. 1.2** Gough platform (1947) reproduced here with the kind permission of Dunlop Tyres Co. The moving platform, to which the tire is attached, is linked to the ground by six legs consisting each of a universal joint, an actuated prismatic joint, and a spherical joint arranged in series. The actuation of the prismatic joints modifies the position and orientation of the moving platform, and therefore of the wheel. The wheel is driven by a conveyor belt and the mechanism permits measuring the tire wear under varying conditions

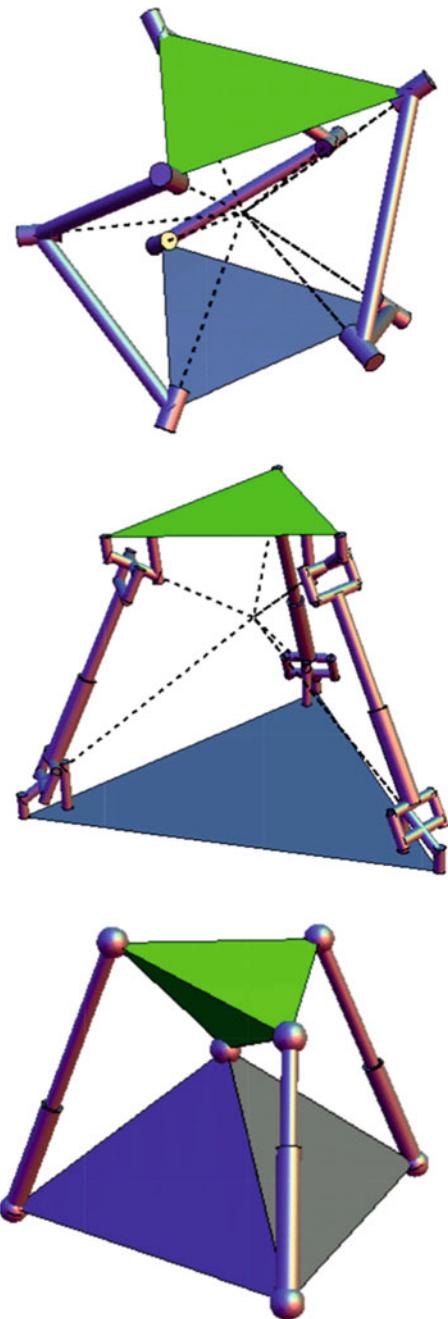


**Fig. 1.3** 3RRR spherical parallel robot sometimes call *agile eye* because of its common use as a three-degree-of-freedom camera-orienting device (drawings adapted from [26])

In each leg of a Gough–Stewart platform, the centers of the spherical joints are called *attachment points*, which are fixed either to the base or the moving platform, whose distance (leg length) is controlled by the actuated prismatic joint. Two or more attachment points, either in the base or in the moving platform, can coalesce into a unique point. According to the number of attachment points (no matter if they are multiple or not) in the moving platform, say  $p$ , and in the fixed base, say  $q$ , different architectures, named  $p-q$ , are distinguished [21].

The most popular spherical fully parallel robot is probably the 3RRR robot in Fig. 1.3, schematically represented in Fig. 1.4 (top), in which each leg is composed of three revolute joints. The axes of the nine revolute joints pass through the center of the spherical motion. The 3UPU robot in Fig. 1.4 (center) also behaves as a spherical robot when the revolute axes of the universal joints are properly arranged [13]. Nevertheless, the spherical counterpart of the Gough–Stewart platform is considered to be the S-3SPS robot [see Fig. 1.4 (bottom)] where the moving platform is articulated with respect to the fixed base through a passive spherical joint and three SPS legs connecting the base and the platform in parallel [10].

Throughout this book, we will use as a reference of spatial and spherical parallel robots the 6SPS robot and the S-3SPS robot, respectively. In other words, we will limit our study to parallel robots with SPS legs. Since these legs contain an idle motion (the rotation about the axis defined by the centers of the two spherical joints), they can be substituted with UPS or SPU legs without affecting the motion capabilities of the moving platform, which is a common practice in most implementations. The universal joints can also be substituted with two consecutive R joints, in which case we can have either robots with RRPS or SPRR legs, a substitution that will be useful when using lockable joints.



**Fig. 1.4** Three different parallel spherical robots. From top to bottom: the 3RRR robot, the 3UPU robot, and the S-3SPS robot. The last one is commonly considered as the spherical counterpart of the Gough–Stewart platform because it features three SPS legs

Fully-parallel robots are used in applications where accuracy, stiffness or high speeds and accelerations are required [29]. However, one of their main drawbacks are their small workspaces compared to their serial counterparts. This is due, in part, to the existence of potential collisions between the different elements of the robot and the working ranges of the joints. Another feature which seriously reduces the workspace of fully-parallel robots is the existence of singularities thus highly complicating the moving platform location control unless the workspace is artificially reduced to leave these singularities out of it.

In this book, we propose new parallel robots which can be seen as parallel robots with SPS legs where some of their passive joints have been replaced with lockable or non-holonomic joints. We will see how these substitutions allow reducing the number of legs, and hence the number of actuators, without reducing the dimension of the robot's workspace. Reducing the number of actuators has important consequences including a reduction in robot's weight and cost, and in the possibility of reducing the possible collisions between legs. This opens the opportunity of enlarging the robot's workspace. Unfortunately, we will see how this reduction in the number of actuators makes these new robots unable to follow arbitrary trajectories in their configuration spaces, thereby increasing the complexity of their motion planning algorithms as they have to perform, in general, maneuvers to reach the desired pose.

The impossibility of following arbitrary trajectories is not a problem for many applications. Robots usually require a high positioning accuracy in some locations defined by the task to be performed, while the path connecting them can run inside predetermined margins. For example, in pick-and-place tasks, while high accuracies are needed in the initial and final configurations, the exact trajectory followed by the robot is unimportant, or is limited to a wide range. The ideal parallel robot for these tasks should be able to make the moving platform reach any pose in the six-dimensional workspace, and, by exploiting the free fly of the moving platform, it should be able to satisfy additional design conditions that reduce its hardware complexity. In general, the reachable relative poses of two links connected by lockable or non-holonomic joints is not constrained with respect to their standard counterparts, they have the only effect of reducing the set of paths that can be followed from a given configuration. This reduction of practicable paths can be usefully exploited to eliminate actuators. Thus, a robot with fewer actuators than the dimension of its configuration space—to reduce bulk, weight and expense—becomes feasible by introducing unconventional joints such as the ones considered in this book.

We will see how replacing ordinary joints with lockable or non-holonomic joints leads to on-line reconfigurable or underactuated robots, respectively. In both cases, the resulting robots have, in general, a larger workspace at a lower cost. The main price to pay is a reduction in speed as maneuvers have to be introduced to either approximate arbitrary trajectories or to reach arbitrary configurations. The design of the motion planning algorithms to automatically generate these maneuvers is the major challenge faced during the development of these robots.

To better understand the complexity of the problem, next we briefly introduce the kind of robots generated by the aforementioned substitutions.

### ***1.1.1 Reconfigurable Robots with Lockable Joints***

If a leg is removed from a 6RRPS parallel robot, the moving platform and the remaining passive joints are free to move within certain ranges. Then, if one of remaining passive R joints is locked, the system stiffness can be recovered. Obviously the moving platform would have one degree of freedom less. Nevertheless, if instead of only locking one of the remaining passive joints we lock some of them alternatively, the robot's architecture changes. Then, if the lockable joints are properly placed, the union of the workspaces of all possible architectures that can be generated will approximate that of the original parallel robot. In other words, the robot may move in the same workspace as that of the original robot despite having one actuator less. Now, in general, the robot will have to pass through different configurations of locked and released joints to reach a desired configuration. In other words, maneuvers need to be planned to reach arbitrary configurations. In principle, these maneuvers could also be useful to avoid singularities.

A lockable revolute joint, which we will denote by  $R_b$ , can be seen as a standard R joint plus a binary actuator (usually implemented using a clutch) that enables or disables the mobility of the joint at will. Now, consider the series connection of a lockable revolute joint and a ordinary revolute joint whose axes are perpendicular and intersecting in a point as shown in Fig. 1.5. The first joint is lockable which can be implemented using an electromechanical clutch, and the second joint is passive. As a consequence, this joint behaves as a U joint if the clutch is disengaged, or as a R joint if it is engaged.

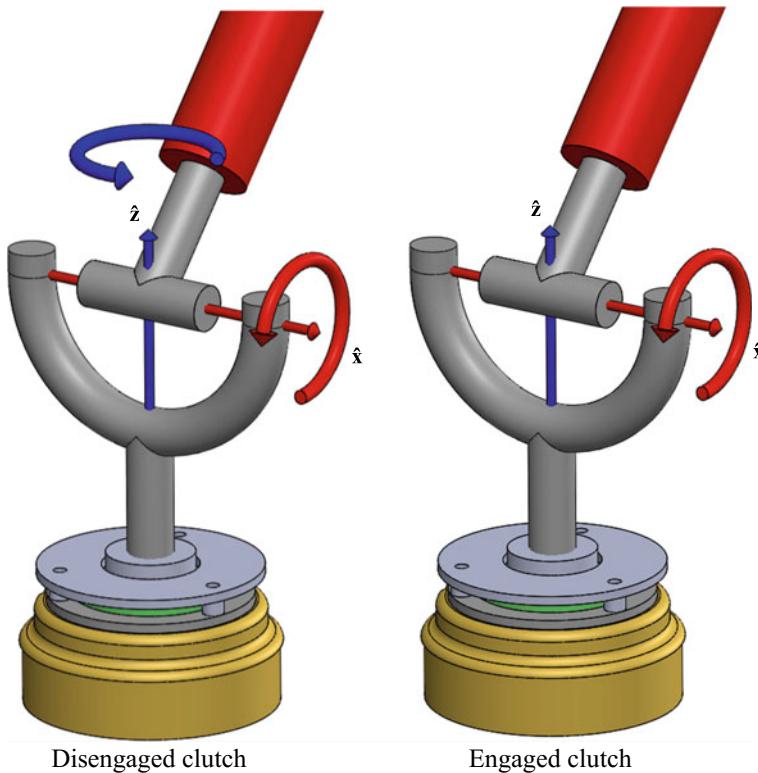
Chapter 2 of this book is devoted to the study of the parallel robots resulting from replacing the U joints in UPS legs by the aforementioned  $R_b$  set of joints. This will permit reducing the number of legs. At least one  $R_b$  per each leg substitution is needed to keep the robot away from collapsing, but more are needed to enable full mobility using switching sequences.

### ***1.1.2 Underactuated Robots with Non-holonomic Joints***

Underactuation is a term used in robotics to describe robots that cannot be commanded to follow arbitrary trajectories in configuration space. This condition can occur for a number of reasons, being the presence of non-holonomic constraints the most common one.

Let us consider the non-holonomic spherical joint shown in Fig. 1.6. It can be seen as a standard spherical joint where a cylinder, free to rotate about its axis, is put in contact with the sphere. According to the chosen reference frame centered on the sphere, the cylinder is placed so that its rotating axis is parallel to the  $x$ -axis, and the contact point with the sphere intersects the  $y$ -axis. Then, the sphere

- can rotate around the  $x$ -axis. As the sphere rotates about the  $x$ -axis, a rotation of the cylinder is induced due to friction forces at the contact point.

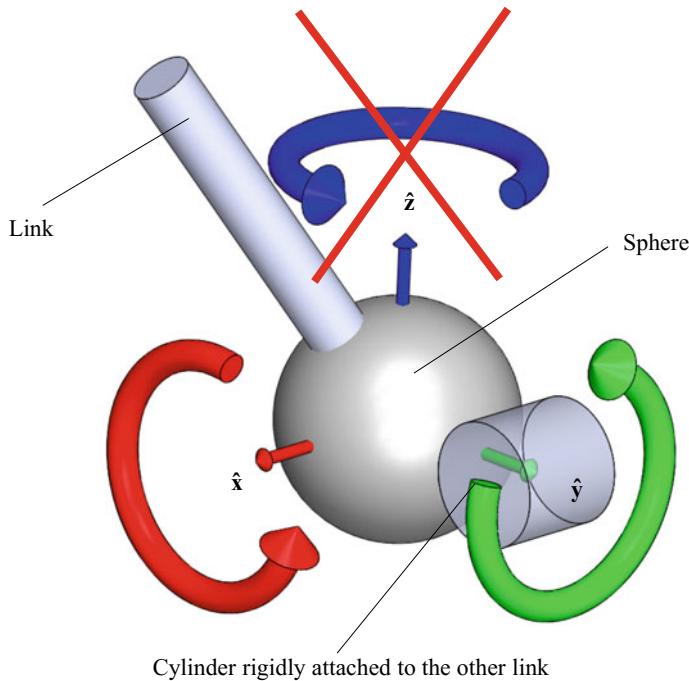


**Fig. 1.5** The series connection of a lockable revolute joint, implemented using a clutch, and a ordinary revolute joint behaves as a universal joint if the clutch is disengaged (left), or as a revolute joint if it is engaged (right)

- can rotate around the  $y$ -axis, pivoting around the contact point. As no torque is generated on the cylinder axis, its orientation remains unaltered.
- cannot rotate about the  $z$ -axis. Since the cylinder cannot rotate about the  $z$ -axis, the friction forces at the contact point prevents the rotation of the sphere about this axis.

Although the resulting *non-slip* contact only allows the sphere to rotate about the  $x$ - or the  $y$ -axis, by combining both rotations, the sphere can attain an equivalent rotation about the  $z$ -axis. For example, a rotation of  $\pi/2$  radians about the  $x$ -axis, followed by a rotation of  $\theta$  radians about  $y$ -axis, and finally a rotation of  $-\pi/2$  radians about the  $x$ -axis is clearly equivalent to a rotation of  $\theta$  radians about the  $z$ -axis.

If a conventional spherical joint in a parallel robot is substituted with such a joint, the possible orientations between the two links this joint connects remain unconstrained, only the relative velocities between them are limited because of this substitution. In other words, only the trajectories connecting two arbitrary orientations



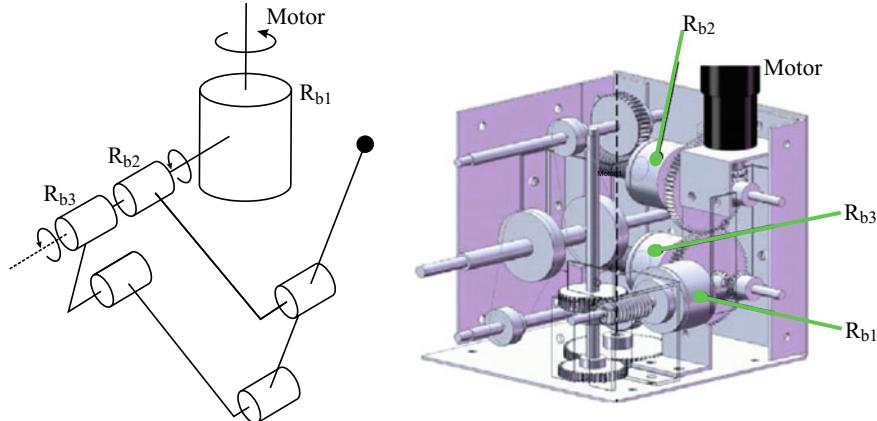
**Fig. 1.6** Non-holonomic spherical joint

between the two connected bodies are constrained. This non-holonomic constrain also introduces reaction forces that can be used to reduce the number of actuators, thus allowing designs having less actuators than the dimension of their configuration spaces.

Chapters 4 and 5 of this book are devoted to the study of the parallel robots resulting from replacing the spherical joint in their SPU legs with the aforementioned non-holonomic joint. This permits reducing the number of legs, one per each substitution.

## 1.2 Precursors

To the best of our knowledge, at the beginning of the development of this research, there were no previous results on the use of lockable and non-holonomic joints in parallel robots. Then, in absence of directly connected previous works, next we summarize those ones that have been used, as a source of inspiration, during the development of this work.



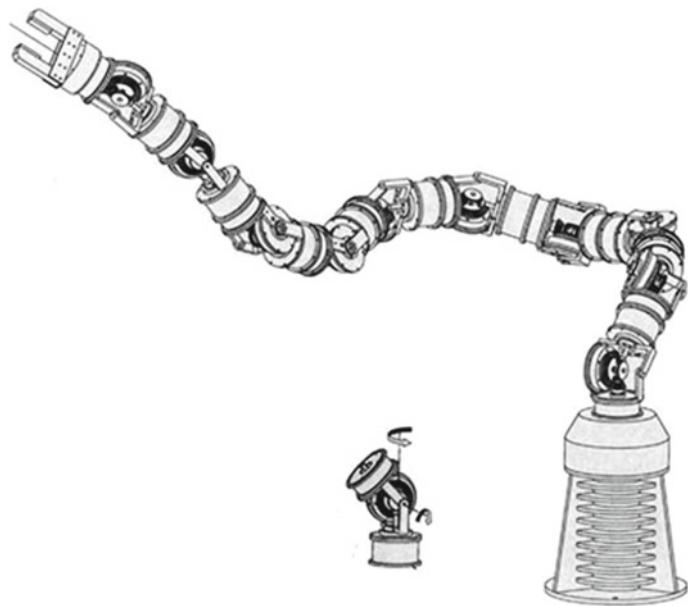
**Fig. 1.7** Clutched-arm robot with three lockable revolute joints and only one motor. Although classified as a serial robot, strictly speaking it is a serial-parallel mechanism. Left: conceptual design. Right: Mechanical design. Images reproduced here with the kind permission of Marco Cecarelli from the University of Cassino, Italy

### 1.2.1 Precursors of Robots with Lockable Joints

The use of lockable joints in robot designs is not common. A very small number of examples can be found in the literature which can be categorized into the two groups described below.

One group uses several revolute joints that can be engaged with a single motor. While not engaged, they remain blocked. Then, the state of the joints can be switched between locked and actuated. It is a good strategy to meet the design requirements of minimum weight by minimizing the number of motors and the battery size. Within this group, we can find serial robots like the LARM clutched-arm described by Gu and Ceccarelli in [17, 18] (see Fig. 1.7), or like the hyper-redundant snake-like robot presented by Zhu in [39] (see Fig. 1.8), which uses gears for transmission between modules. Another two examples of hyper-redundant serial robots with lockable revolute joints are the uni-drive modular robot [22] and the 3D-Trunk [31]. The first one, in contrast to the one presented by Zhu, uses a flexible transmission shaft with clutches at each point of actuation as shown in Fig. 1.9 (top). The second one, depicted in Fig. 1.9 (bottom), exploits the use of two pairs of opposed tendons to generate torsion in all joints and, by unlocking one clutch, to generate a rotation at the corresponding joint.

The other group is represented by the serial robots using prismatic lockable joints. These joints, denoted as  $P_b$  joints, were first proposed by Aghili and Parsa [1, 2]. They bring two main advantages: it is possible to on-line expand the robot's workspace (see Fig. 1.10), and to reduce the robot's dimensions for transportation.



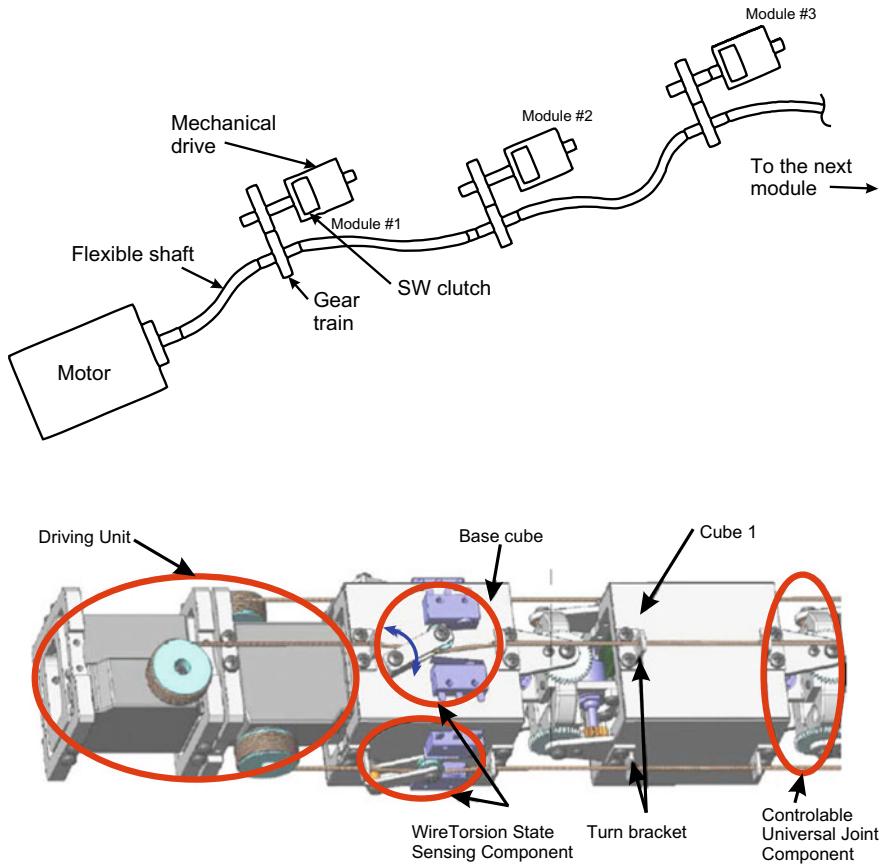
**Fig. 1.8** Example of serial robot with lockable revolute joints and only one motor. Modular 20-joints hyper-redundant serial robot propose by Zhu [39]

### 1.2.2 Precursors of Robots with Non-holonomic Joints

We have just seen how designing a robot with fewer actuators than the dimension of its configuration space—to reduce bulk, weight and cost—becomes feasible by introducing lockable joints. Now, we will show how this is also possible by introducing non-holonomic mechanical elements. Unfortunately, as with lockable joints, the mechanical advantages of these non-holonomic designs are usually darkened by the complexity of their control. Considerable effort has been made to clarify different aspects of non-holonomic mechanical systems [5]. A challenge in control of these systems results from a limited applicability of the feedback control, discovered by Brockett [8] and Lizárraga [25].

Non-holonomic joints, a mechanical concept probably used for the first time in [23], can be implemented using convex bodies rolling on spherical surfaces. Two kinds of contacts can be considered: marble rolling, when the convex body can freely roll in contact with the sphere without slipping [7], and rubber rolling, when the convex body satisfies additionally a no-twist condition [24].

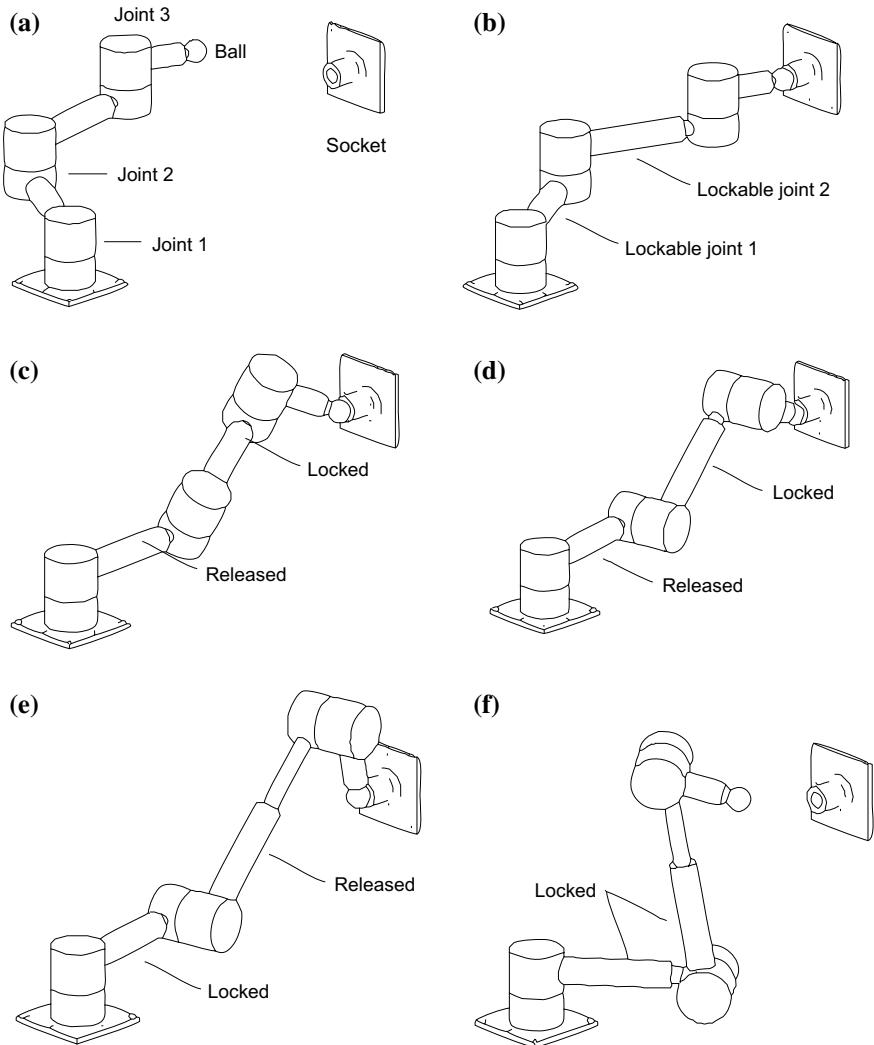
In the practical implementations of non-holonomic joints, the rolling convex body is usually a disk implementing a marble rolling contact with the sphere. If the disk rolls upon the interior surface of a spherical shell, the resulting joint is said to implement the Suslov constraint [38]. Alternatively, if the contact is performed on the outer



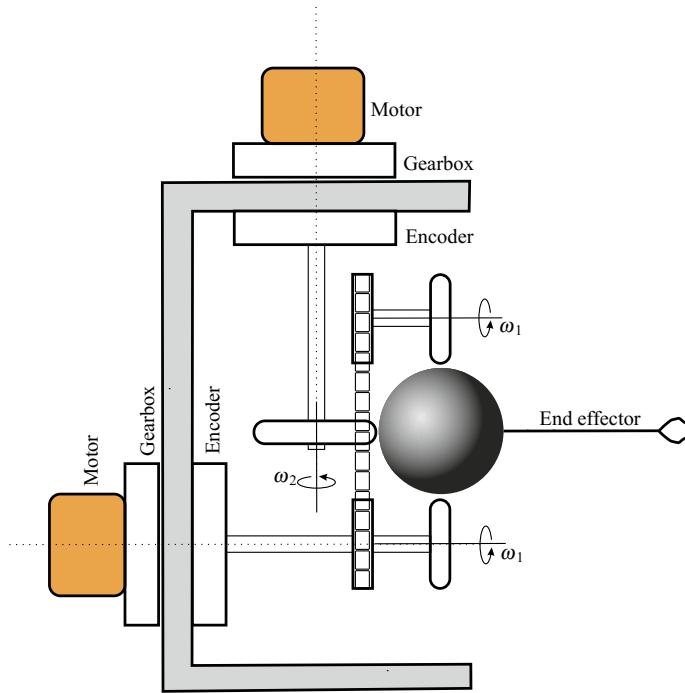
**Fig. 1.9** Examples of two serial robots with lockable revolute joints and only one motor. Top: schematics of the uni-drive modular robot propose by Karbasi. Bottom: 3D-Trunk wire-drive system proposed by Ning and Worgötter. Images adapted from [22, 31], respectively

surface of the spherical shell, the resulting joint is said to implement the Veselova constraint [6], the kind of non-holonomic joint used throughout this book.

Lower-mobility spatial parallel robots have become an active research topic in the field of parallel robot during the last decade because of their simple structure, low price and easy control. The dimension of the space of admissible velocities for the end-effector of this kind of parallel robots is lower than six and, if singular configurations are excluded, equal to the dimension of the tangent space of the reachable manifold. The substitution of a standard joint in a lower-mobility parallel robot with a non-holonomic joint with equivalent instantaneous kinematics has dramatic consequences: while the dimension of the space of admissible velocities for the end-effector remains the same, the dimension of the reachable space is increased. To the best of our knowledge, this idea was first used by Ben-Horin and Thomas in



**Fig. 1.10** Maneuver for the RP<sub>b</sub>RP<sub>b</sub>R serial robot proposed by Aghili and Parsa to modify the lengths of its lockable prismatic joints (adapted from [2]). It can actually be seen as a reconfigurable RRR serial robot

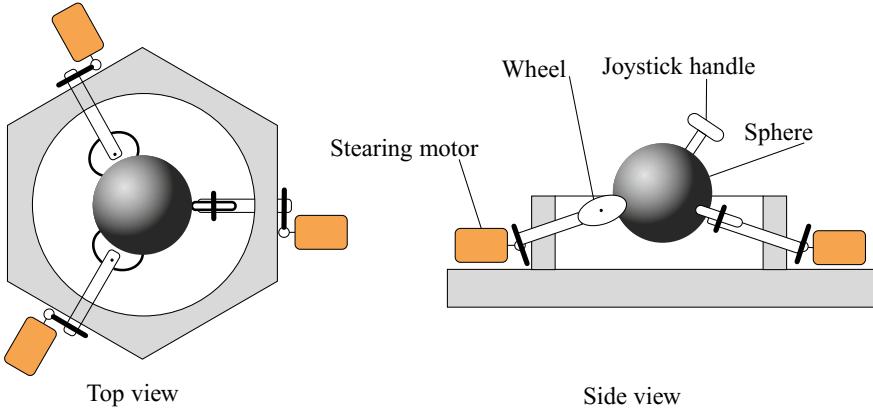


**Fig. 1.11** The non-holonomic robotic wrist proposed by Stammers et al. (adapted from [36])

[4], where a three-legged parallel robot is proposed, where each leg is connected to the base through a non-holonomic joint.

Non-holonomic constraints appear in multiple areas of robotics [12, 14, 20, 35], mainly in those related to mobile robots (either terrestrial, outer-space, or underwater vehicles), or the manipulation of objects with multiple contacts. In all of these cases the non-holonomic restrictions are inherent. Angeles in [3], O'Reilly in [32], and more particularly Hennessey in [19], give a general view on how to deal with problems involving non-holonomic constraints. In particular, Hennessey presents an analysis of a unicycle moving on a sphere, an example closely related to the non-holonomic spherical joint used in this book. The difference is that in our case the moving element is the sphere with respect to a fixed base and one of the control inputs affect only one of the state variables.

The literature on the use of non-holonomic devices in the design of robots is limited to few examples. For example, in [36], Stammers et al. present a robot wrist that can attain any orientation with only two motors. This is achieved by means of a friction drive, using rollers on a spherical ball to which the end effector is fixed, and by fixing the two motors to the arm. The roller axes are perpendicular to each other, which means that rotate instantaneously around the normal of the plane defined by them is not possible except by composition of rotations (Fig. 1.11).

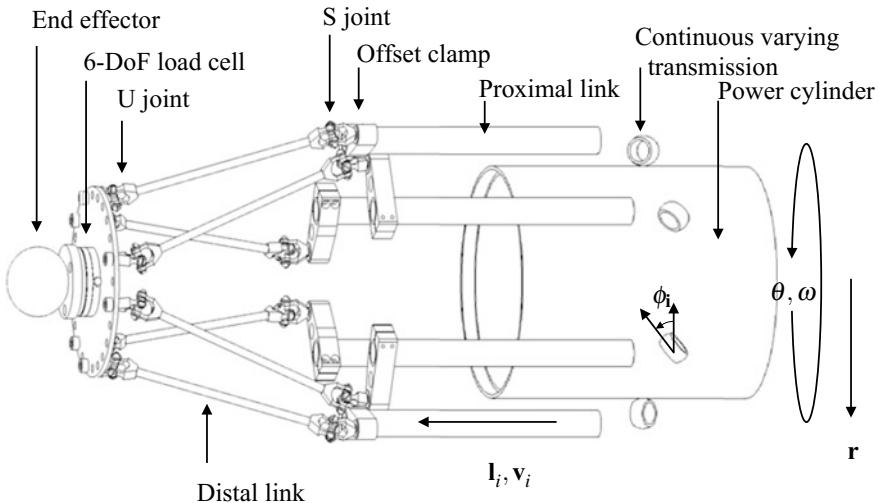


**Fig. 1.12** Conceptual design of a spherical cobot based on non-holonomic contacts (adapted from [33])

Peshkin et al. [33] presented a group of haptic devices, named cobots, with programmable constraints generated by non-holonomic joints. The non-holonomic constraints are introduced by rollers moving over different surfaces. The axis of rotation of each roller is adjustable, the axis is normal tangent plane at the contact point surface-roller. Each cobot differ in the type of surface used. For example, the spherical cobot consist of a sphere with a joystick resting on three rollers (Fig. 1.12), and the planar cobot is a single roller over a plane. The most interesting cobot, from the point of view of the work developed in this book, is the one with architecture 6PSU, which uses rollers in contact with a continuously rolling cylinder to actuate the six prismatic joints. The rollers in contact with the cylinder implement non-holonomic joints used as continuous varying transmissions (Fig. 1.13).

Nakamura et al. [30] proposed a serial robot with  $n$  joints which can reach any pose in its  $n$ -dimensional configuration space with only two actuators. The joints of this robot are coupled by non-holonomic devices, based on spheres and rollers, so that it reaches a desired pose by following a path whose computation is algorithmically equivalent to maneuvering a car with  $n$ -trailers (see Fig. 1.14).

More recently, Ben-Horin and Thomas proposed a three-legged parallel robot where each leg is connected to the base through a sphere whose motion is constrained by a roller [4]. This parallel architecture permits to attain any position and orientation for the platform using only three prismatic actuators. It can be said that this work triggered our interest in the use of non-holonomic joints in parallel robots.



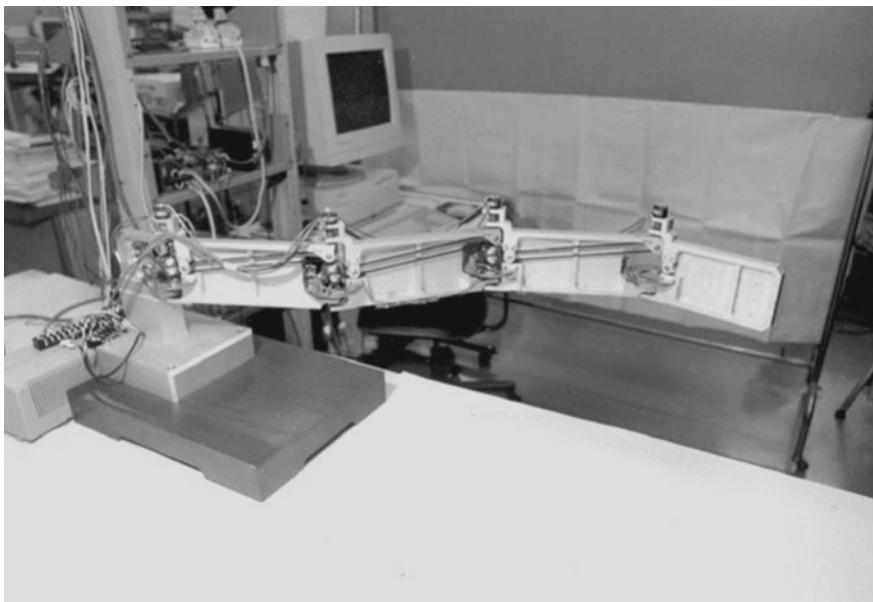
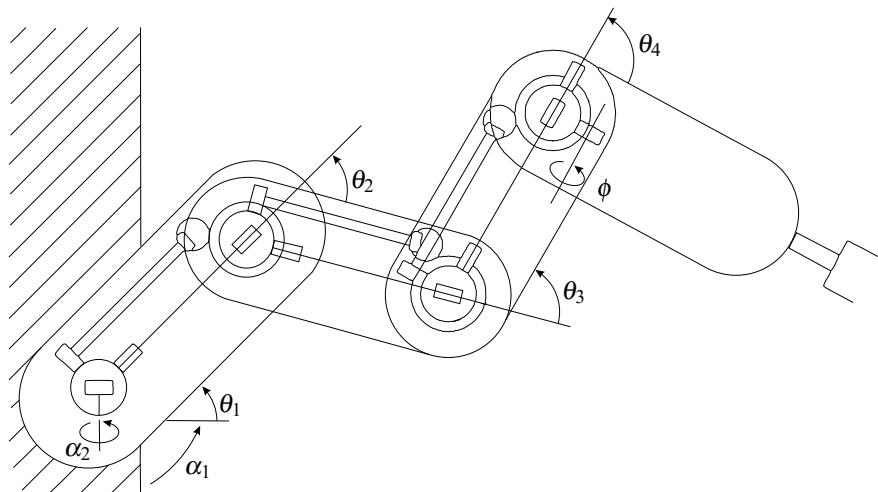
**Fig. 1.13** Conceptual design of a 6-degrees-of-freedom cobot haptic device based on non-holonomic joints (adapted from [15])

### 1.3 Organization of this Book

Chapter 2 is fully devoted to parallel robots with revolute lockable joints actuated by clutches. Due to its symmetry and because it uses the least number of actuators, the effort is focused in the analysis and implementation of a parallel robot consisting of four R<sub>b</sub>RPS legs.

While the study of other designs of parallel robots involving the considered lockable joint will closely follow the one presented in Chap. 2, the situation is much more complicated when working with parallel robots having non-holonomic joints. This is why the following five chapters of this book are devoted to parallel robots with this kind of joints, analyzing separately the spatial and the spherical case.

Chapter 3 deepens in the aforementioned idea of generating under-actuated parallel robots by substituting spherical joints with non-holonomic spherical joints. Particular attention is also paid in this chapter to the practical implementation of non-holonomic joints. In Chap. 4, the kinetostatic analysis of the 3S<sub>n</sub>PU under-actuated spatial parallel robot is presented. This analysis is then used to develop the path planner presented in Chap. 5. The spherical case deserves a separate study due to the simplifications that can be introduced with respect to the spatial case. The study of the S<sub>n</sub>-2SPU under-actuated spherical parallel robot is done in Chaps. 6 and 7. In Chap. 6, first the kinematics and then the workspace of the robot are analyzed. In Chap. 7, different path planners are developed, including one that can be used as part of a controller. Finally, Chap. 8 summarizes the main points presented in the book and gives some prospects for future research.



**Fig. 1.14** Conceptual design of the four-joint robot with only two actuators proposed by Nakamura et al. (top) and its practical implementation (bottom)

## References

1. F. Aghili, K. Parsa, Design of a reconfigurable space robot with lockable telescopic joints, in *IEEE/RSJ International Conference on Intelligent Robots and Systems* (2006), pp. 4608–4614
2. F. Aghili, K. Parsa, A reconfigurable robot with lockable cylindrical joints. *IEEE Trans. Robot.* **25**(4), 785–797 (2009)
3. J. Angeles, *Fundamentals of Robotic Mechanical Systems: Theory, Methods, and Algorithms*, Mechanical Engineering Series (2003)
4. P. Ben-Horin, F. Thomas, A nonholonomic 3-DoF parallel robot, in *Advances in Robot Kinematics: Analysis and Design* (Springer, Berlin, 2008), pp. 111–118
5. A.M. Bloch, *Nonholonomic Mechanics and Control*, Interdisciplinary Applied Mathematics, vol. 24 (2003)
6. A.V. Borisov, I.S. Mamaev, Chaplygin’s Ball. The Suslov problem and Veselova’s problem. Integrability and realization of constraints, in *Nonholonomic Dynamical Systems. Integrability, Chaos, Strange Attractors* (Institute of Computer Science, Izhevsk, Moscow, 2002), pp. 118–130
7. A.V. Borisov, I.S. Mamaev, Rolling of a rigid body on plane and sphere. Hierarchy of dynamics. *Regul. Chaotic Dyn.* **7**(2), 177–200 (2002b)
8. R.W. Brockett, Asymptotic Stability and Feedback Stabilization, in *Differential Geometric Control Theory* (Birkhauser, Boston, 1983), pp. 181–191
9. J. Cortés, T. Siméon, Probabilistic motion planning for parallel mechanisms, in *IEEE International Conference on Robotics and Automation*, Taipei, Taiwan (2003), pp. 4354–4359
10. G. Cui, Y. Zhang, Kinetostatic modeling and analysis of a new 3-DoF parallel manipulator, in *IEEE International Conference on Computational Intelligence and Software Engineering*, Wuhan, China (2009), pp. 1–4
11. B. Dasgupta, T.S. Mruthyunjaya, The stewart platform manipulator: a review. *Mech. Mach. Theory* **35**(1), 15–40 (2000)
12. A. De Luca, G. Oriolo, P. Robuffo-Giordano, Image-based visual servoing schemes for non-holonomic mobile manipulators. *Robotica* **25**(2), 131–145 (2007)
13. R. Di Gregorio, Statics and singularity loci of the 3-UPU wrist. *IEEE Trans. Robot.* **20**(4), 630–635 (2004). ISSN 1941-0468
14. I. Duleba, W. Khefifi, Velocity space approach to motion planning of nonholonomic systems. *Robotica* **25**(3), 359–366 (2007)
15. E. Faulring, J. Colgate, M. Peshkin, A high performance 6-DoF haptic cobot, in *IEEE International Conference on Robotics and Automation*, New Orleans, LA, USA, vol. 2, no. 2 (2004), pp. 1980–1985
16. V.E. Gough, S.G. Whitehall, Universal Tyre Test Machine, in *International Automobile Technical Congress. Proceedings of the FISITA Ninth Institution of Mechanical Engineers*, ed. by G. Eley (1962), pp. 117–137
17. H. Gu, M. Ceccarelli, Simulation of combined motions for a 1-DoF clutched robotic arm, in *International Conference on Mechatronics and Automation*, Changchun, China (2009), pp. 3721–3726
18. H. Gu, N.E. Nava, M. Ceccarelli, Dynamics simulation of operation for a clutched arm, in *XVIII Congreso Nacional de Ingeniera Menica,Ciudad Real, Espaa* (2010), pp. 1342–1349
19. M.P. Hennessey, Visualizing the motion of a unicycle on a sphere. *Int. J. Model. Simul.* **26**(1), 69–79 (2006)
20. I.I. Hussein, A.M. Bloch, Optimal control of underactuated nonholonomic mechanical systems. *IEEE Trans. Autom. Control* **53**(3), 668–682 (2008)
21. C. Innocenti, V. Parenti-Castelli, Exhaustive enumeration of fully parallel kinematic chains, in *Proceedings ASME International Winter Annual Meeting*, Chicago, IL, USA, vol. 55, no. 2 (1994), pp. 1135–1141
22. H. Karbasi, J.P. Huissoon, A. Khajepour, Uni-drive modular robots: theory, design, and experiments. *Mech. Mach. Theory* **39**(2), 183–200 (2004)

23. A.P. Kharlamov, M.P. Kharlamov, Nonholonomic joint, in *Mekh. Tverd. Tela (Rigid Body Mechanics)*, vol. 27 (NAS of Ukraine, Ukraine, 1995), pp. 1–7
24. J. Koiller, K. Ehlers, Rubber rolling over a sphere. *Regul. Chaotic Dyn.* **12**(2), 127–152 (2007)
25. D.A. Lizárraga, Obstructions to the existence of universal stabilizers for smooth control systems. *Math. Control Signals Syst. (MCSS)* **16**(4), 255–277 (2004)
26. S. Lohuis, M. Abdelhary, S. Abdelmoeti, T. de Vries, Demo design: the VIRO agile eye. *Mikroniek* **58**(1), 10–16 (2018)
27. Y. Lu, J. Xu, Simulation solving/modifying velocity and acceleration of a 4UPS+SPR type parallel machine tool during normal machining of a 3D free-form surface. *Int. J. Adv. Manuf. Technol.* **42**(7–8), 804–812 (2009)
28. J.P. Merlet, Singular configurations of parallel manipulators and grassmann geometry. *Int. J. Robot. Res.* **8**(5), 45–56 (1989)
29. J.P. Merlet, Les Robots Parallèles. *Trait des nouvelles technologies - Robotique* (1997)
30. Y. Nakamura, W. Chung, O.J. Srdalen, Design and control of the nonholonomic manipulator. *IEEE Trans. Robot. Autom.* **17**(1), 48–59 (2001)
31. K. Ning, F. Worgotter, A novel concept for building a hyper-redundant chain robot. *IEEE Trans. Robot.* **25**(6), 1237–1248 (2009)
32. O.M. O'Reilly, *Intermediate Dynamics for Engineers* (2008)
33. M. Peshkin, J.E. Colgate, C. Moore, Passive robots and haptic displays based on nonholonomic elements, in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 1 (1996), pp. 551–556
34. F. Pierrot, P. Dauchez, A. Fournier, HEXA: a fast six-DoF fully-parallel robot, in *Fifth International Conference on Advanced Robotics 'Robots in Unstructured Environments*, vol. 2 (1991), pp. 1158–1163
35. E.A. Shamma, H. Choset, A.A. Rizzi, Towards a unified approach to motion planning for dynamic underactuated mechanical systems with non-holonomic constraints. *Int. J. Robot. Res.* **26**(10), 1075–1124 (2007)
36. C.W. Stammers, P.H. Prest, C.G. Mobley, A friction drive robot wrist: electronic and control requirements. *Mechatronics* **2**(4), 391–401 (1992)
37. D. Stewart, A platform with six degrees of freedom. *Proc. Inst. Mech. Eng.* **180**(1), 371–378 (1965)
38. V. Vagner, Geometrical interpretation of the motion of non-holonomic dynamical systems. *Proc. Semin. Vector Tensor Anal.* **5**, 301–327 (1941)
39. H. Zhu, Single motor driven hyper-redundant manipulator (2003). <http://www.imdl.gatech.edu/haihong/Arm/Arm.html>

# Chapter 2

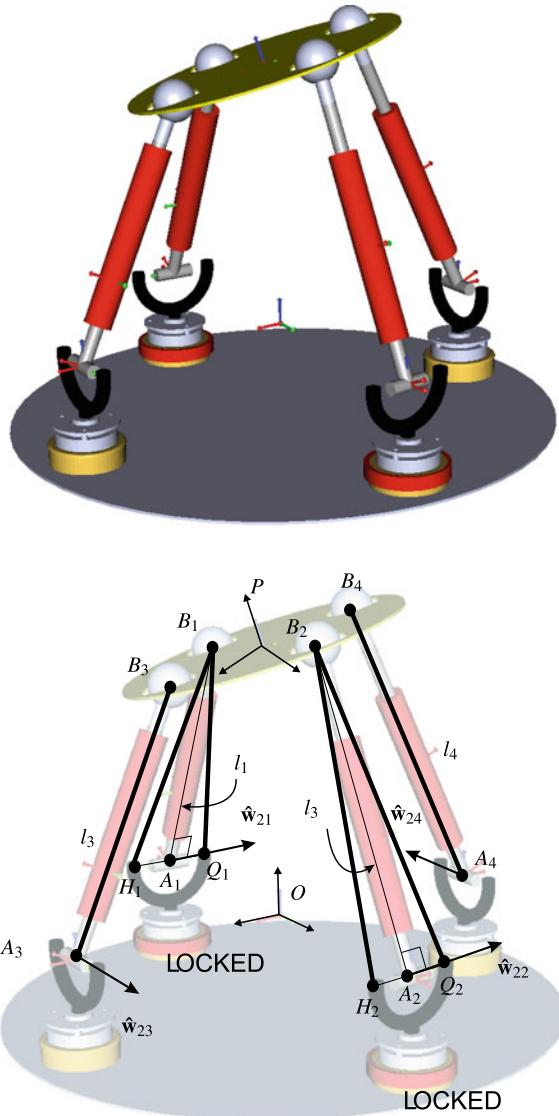
## Parallel Robots with Lockable Revolute Joints



This chapter describes a class of reconfigurable parallel robots consisting of a fixed base and a moving platform connected by  $n R_b R_{PS}$  serial chains, with  $n < 6$ . Only the prismatic joints are actuated and the first revolute joint in each chain can be locked or released during operation. It will be shown how the introduction of these lockable joints allow the prismatic actuators to maneuver to approximate 6-degrees-of-freedom motions for the moving platform. An algorithm for generating these maneuvers is also described. Then, a motion planner, based on the generation of a probabilistic road map, whose nodes are connected using the described maneuvers, is presented. The generated trajectories are also designed to avoid both singularities and possible collisions between legs. Part of the work presented in this chapter appeared in [8].

As we have already seen, the Gough–Stewart platform consists of a base and a moving platform connected by six  $UPS$  legs. If some of these  $UPS$  legs are eliminated, the moving platform will have the same number of degrees of freedom as the number of eliminated legs. Some of the remaining passive joints must be blocked to keep the moving platform rigidly linked to the base. We can substitute all the remaining legs with  $R_b R_{PS}$  legs, where  $R_b$  denotes the lockable revolute joint described in the previous chapter. To keep the moving platform location controllable at all times, the number of locked  $R_b$  must be equal to the number of eliminated legs. Each  $R_b R_{PS}$  leg will behave either as a  $R_{PS}$  chain when the  $R_b$  joint is locked, or as a  $UPS$  when it is released.

Table 2.1 summarizes the family of generated robots after performing different numbers of substitutions proceeding as explained above. Observe that the maximum number of leg eliminations is three, but in this case the three lockable joints in the final design must be locked at all times so this case does not deserve further attention. The  $5R_b R_{PS}$  and the  $4R_b R_{PS}$  architectures are the only ones of interest because their motion possibilities can be increased by on-line switching their lockable joints. Two kinds of new reconfigurable parallel robots are thus obtained. The architecture involving four legs is probably the most attractive because it uses the least number of actuators (Fig. 2.1). This chapter is devoted to its study.



**Fig. 2.1** The 4R<sub>b</sub>RPS parallel robot and its associated notation. The four legs are attached to the base through lockable revolute joints, and to the moving platform through spherical joints. Since two clutches must be engaged at any time to keep the moving platform rigidly linked to the base, it behaves as a reconfigurable 2RPS-2UPS robot

**Table 2.1** The three possible architectures resulting from replacing some UPS legs of a Gough-Stewart platform with R<sub>b</sub>RPS legs

Legs removed	Equivalent reconfigurable architecture	Dimension of the workspace	Related references
1	4UPS-RPS	5 (6 if on-line reconfigured)	[18]
2	2UPS-2RPS	4 (6 if on-line reconfigured)	[10, 16, 17]
3	3RPS	3	[11, 20]

This chapter is organized as follows. Section 2.1 studies the kinematics of the proposed robot. Section 2.2 shows how to maneuver to locate the moving platform in any arbitrary pose. Section 2.3 shows how to generate a road map in the configuration space of the robot that permits to obtain paths, far from singularities and leg collisions, connecting two arbitrary poses. Finally, Sects. 2.4 and 2.5 describe practical aspects concerning the implemented prototype.

## 2.1 Kinematics of the 4R<sub>b</sub>RPS Parallel Robot

Without loss of generality, if the leg lengths of the 4R<sub>b</sub>RPS robot in Fig. 2.2 are fixed and the clutches of legs 1 and 2 are locked, the resulting parallel structure with 2RS-2US topology. The attachments of the legs to the base (moving platform) are represented by points  $A_i$  ( $B_i$ ), for  $i = 1, \dots, 4$ , whose position vectors with respect to the base reference frame is denoted by  $\mathbf{a}_i$  ( $\mathbf{b}_i$ ). The length of leg  $i$ ,  $\|\mathbf{b}_i - \mathbf{a}_i\|$ , is denoted by  $l_i$ . Then,

$$\hat{\mathbf{g}}_i = \frac{(\mathbf{b}_i - \mathbf{a}_i)}{l_i} \quad i = 1, \dots, 4, \quad (2.1)$$

are unit vectors in the direction of the legs. Moreover, according to Fig. 2.2, the two revolute joint axes of leg  $i$  are denoted by  $\hat{\mathbf{w}}_{1i}$  and  $\hat{\mathbf{w}}_{2i}$ .

For the sake of simplicity, it is assumed that these attachments form squares of size  $a$  (in the base) and  $b$  (in the moving platform). A more general formulation can be found in [8].

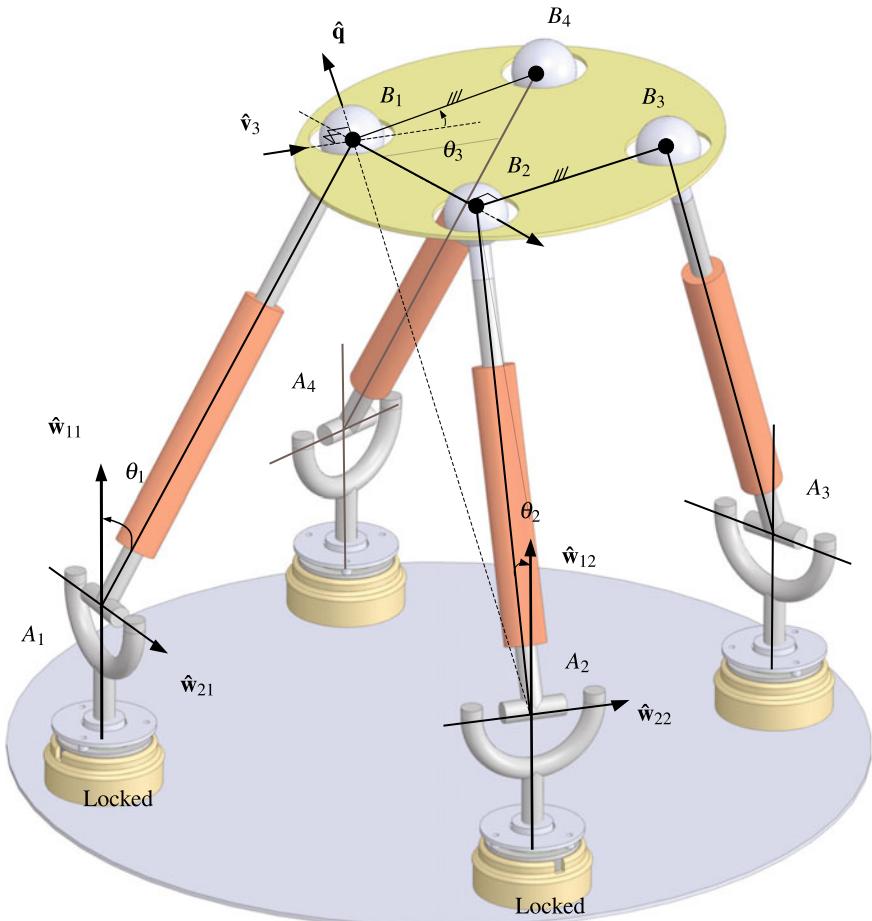
To derive the closure conditions in compact form, we also need to introduce the following unit vectors:

$$\hat{\mathbf{u}}_3 = \frac{\mathbf{b}_2 - \mathbf{b}_1}{b}, \quad (2.2)$$

$$\hat{\mathbf{q}} = \frac{\mathbf{b}_1 - \mathbf{a}_2}{\|\mathbf{b}_1 - \mathbf{a}_2\|}, \quad (2.3)$$

$$\hat{\mathbf{v}}_3 = \frac{\hat{\mathbf{q}} \times \hat{\mathbf{u}}_3}{\|\hat{\mathbf{q}} \times \hat{\mathbf{u}}_3\|}, \quad (2.4)$$

$$\hat{\mathbf{h}}_3 = \hat{\mathbf{u}}_3 \times \hat{\mathbf{v}}_3. \quad (2.5)$$



**Fig. 2.2** Notation associated with the studied 4R<sub>b</sub>RPS robot. The R<sub>b</sub> joints centered at A<sub>1</sub> and A<sub>2</sub> are assumed to be locked and the other two released

### 2.1.1 Position Analysis

The determination of the actuated-joint variables (leg lengths) for an assigned pose of the moving platform (the inverse kinematics problem) is straightforward. In fact, once  $\mathbf{b}_i$  is known, the length of leg  $i$  can be immediately computed since  $\mathbf{a}_i$  is constant and known. On the contrary, the determination of the moving platform pose for assigned leg lengths (the forward kinematics problem) requires the solution of the system of non-linear equations formed by the 2RS-2US closure equations. This problem coincides with the one encountered when solving the forward kinematics of the 6-4 fully-parallel mechanism [10] since that mechanism generates an 2R-2US structure when the actuated joints are locked. In [10], Innocenti gave the analytical solution of this problem and showed that, in general, up to 32 platform poses may be compatible with an assigned set of leg lengths.

In what follows, the 2RS-2US closure equations will be deduced in a form slightly different from the one reported in [10] which is more appropriate for the singularity analysis given in the following section. With reference to Fig. 2.2 and the adopted notations, the 2RS-2US closure equations can be written as follows:

$$(\mathbf{b}_2 - \mathbf{b}_1) \cdot (\mathbf{b}_2 - \mathbf{b}_1) = b^2, \quad (2.6)$$

$$(\mathbf{b}_3 - \mathbf{a}_3) \cdot (\mathbf{b}_3 - \mathbf{a}_3) = l_3^2, \quad (2.7)$$

$$(\mathbf{b}_4 - \mathbf{a}_4) \cdot (\mathbf{b}_4 - \mathbf{a}_4) = l_4^2, \quad (2.8)$$

where

$$\mathbf{b}_1 = \mathbf{a}_1 + l_1(s_1(\hat{\mathbf{w}}_{11} \times \hat{\mathbf{w}}_{21}) + c_1\hat{\mathbf{w}}_{11}), \quad (2.9)$$

$$\mathbf{b}_2 = \mathbf{a}_2 + l_2(s_2(\hat{\mathbf{w}}_{12} \times \hat{\mathbf{w}}_{22}) + c_2\hat{\mathbf{w}}_{12}), \quad (2.10)$$

$$\mathbf{b}_3 = \mathbf{b}_1 + b(c_3\hat{\mathbf{v}}_3 + s_3\hat{\mathbf{h}}_3), \quad (2.11)$$

$$\mathbf{b}_4 = \mathbf{b}_1 + b\hat{\mathbf{u}}_3 + b(c_3\hat{\mathbf{v}}_3 + s_3\hat{\mathbf{h}}_3), \quad (2.12)$$

where  $c_i$  and  $s_i$ ,  $i = 1, \dots, 3$ , stand for  $\cos \theta_i$  and  $\sin \theta_i$ , respectively.

Equation (2.6) is a linear equation in  $c_1$ ,  $c_2$ ,  $s_1$  and  $s_2$ . It is the closure equation of the RSSR loop. Equations (2.7) and (2.8) are linear in  $c_1$ ,  $c_2$ ,  $c_3$ ,  $s_1$ ,  $s_2$  and  $s_3$ . By eliminating  $c_3$  and  $s_3$  from these equations, the resultant will contain  $c_1$ ,  $c_2$ ,  $s_1$  and  $s_2$  which can be used with Eq. (2.6) for a further elimination which will yield an univariate polynomial equation.

### 2.1.2 Singularities

The configurations where the 2RPS-2UPS robot is movable, even though the actuators are locked, are called *singularities*. At these configurations, the robot is not controllable and some links might stand infinite internal forces. Thus, they must be characterized to be avoided during operation.

When the 2RPS-2UPS robot is at a singularity, the 2RS-2US structure obtained by locking the actuators is singular as well (i.e., the structure is not rigid). Thus, by looking for the 2RS-2US singular geometries, the parallel singularities of the associated 2RPS-2UPS can be found.

When the 2RS-2US structure assumes a singular configuration, the moving platform can perform motions that must fulfill the following velocity equations, deduced by differentiating Eqs. (2.6), (2.7), and (2.8):

$$(\dot{\mathbf{b}}_2 - \dot{\mathbf{b}}_1) \cdot \hat{\mathbf{u}}_3 = 0, \quad (2.13)$$

$$\dot{\mathbf{b}}_3 \cdot \hat{\mathbf{g}}_3 = 0, \quad (2.14)$$

$$\dot{\mathbf{b}}_4 \cdot \hat{\mathbf{g}}_4 = 0, \quad (2.15)$$

where

$$\dot{\mathbf{b}}_1 = \dot{\theta}_1 l_1 (\hat{\mathbf{w}}_{21} \times \hat{\mathbf{g}}_1), \quad (2.16)$$

$$\dot{\mathbf{b}}_2 = \dot{\theta}_2 l_2 (\hat{\mathbf{w}}_{22} \times \hat{\mathbf{g}}_2), \quad (2.17)$$

$$\dot{\mathbf{b}}_3 = \dot{\theta}_1 l_1 (\hat{\mathbf{w}}_{21} \times \hat{\mathbf{g}}_1) + \dot{\theta}_3 (\hat{\mathbf{u}}_3 \times (\mathbf{b}_3 - \mathbf{b}_1)) + b (c_3 \dot{\hat{\mathbf{v}}}_3 + s_3 \dot{\hat{\mathbf{h}}}_3), \quad (2.18)$$

$$\begin{aligned} \dot{\mathbf{b}}_4 &= \dot{\theta}_1 l_1 (\hat{\mathbf{w}}_{21} \times \hat{\mathbf{g}}_1) + (\dot{\theta}_2 l_2 (\hat{\mathbf{w}}_{22} \times \hat{\mathbf{g}}_2) - \dot{\theta}_1 l_1 (\hat{\mathbf{w}}_{21} \times \hat{\mathbf{g}}_1)) \\ &\quad + \dot{\theta}_3 (\hat{\mathbf{u}}_3 \times (\mathbf{b}_4 - \mathbf{b}_2)) + b (c_3 \dot{\hat{\mathbf{v}}}_3 + s_3 \dot{\hat{\mathbf{h}}}_3), \end{aligned} \quad (2.19)$$

and the following velocity equations, deduced by differentiating Eqs. (2.2)–(2.5):

$$\dot{\hat{\mathbf{u}}}_3 = \frac{\dot{\theta}_2 l_2 (\hat{\mathbf{w}}_{22} \times \hat{\mathbf{g}}_2) - \dot{\theta}_1 l_1 (\hat{\mathbf{w}}_{21} \times \hat{\mathbf{g}}_1)}{b}, \quad (2.20)$$

$$\dot{\hat{\mathbf{q}}} = \frac{\dot{\theta}_1 l_1 ((\hat{\mathbf{w}}_{21} \times \hat{\mathbf{g}}_1) - (\hat{\mathbf{q}} \cdot (\hat{\mathbf{w}}_{21} \times \hat{\mathbf{g}}_1)) \hat{\mathbf{q}})}{\|\mathbf{b}_1 - \mathbf{a}_2\|}, \quad (2.21)$$

$$\dot{\hat{\mathbf{v}}}_3 = \frac{\dot{\hat{\mathbf{q}}} \times \hat{\mathbf{u}}_3 + \hat{\mathbf{q}} \times \dot{\hat{\mathbf{u}}}_3 - (\hat{\mathbf{v}}_3 \cdot (\dot{\hat{\mathbf{q}}} \times \hat{\mathbf{u}}_3 + \hat{\mathbf{q}} \times \dot{\hat{\mathbf{u}}}_3)) \hat{\mathbf{v}}_3}{\|\hat{\mathbf{q}} \times \hat{\mathbf{u}}_3\|}, \quad (2.22)$$

$$\dot{\hat{\mathbf{h}}}_3 = \dot{\hat{\mathbf{u}}}_3 \times \hat{\mathbf{v}}_3 + \hat{\mathbf{u}}_3 \times \dot{\hat{\mathbf{v}}}_3. \quad (2.23)$$

Now, the introduction of (2.16) and (2.17) in (2.13) yields

$$\dot{\theta}_2 = \dot{\theta}_1 \frac{l_1 \hat{\mathbf{u}}_3 \cdot (\hat{\mathbf{w}}_{21} \times \hat{\mathbf{g}}_1)}{l_2 \hat{\mathbf{u}}_3 \cdot (\hat{\mathbf{w}}_{22} \times \hat{\mathbf{g}}_2)}. \quad (2.24)$$

In the above relationship,  $\dot{\theta}_2$  is undefined when  $\hat{\mathbf{u}}_3$ ,  $\hat{\mathbf{w}}_{22}$ , and  $\hat{\mathbf{g}}_2$  are coplanar. The configuration where this geometric condition occurs are singularities of the internal RSSR loop and, in general, they are singularities of the 2RS-2US structure, too. The introduction of (2.24) into (2.18)–(2.23) and of the resultant relationships into Eqs. (2.14) and (2.15) yield a linear and homogeneous system of two equations in two unknowns which can be written as follows:

$$\begin{pmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{pmatrix} \begin{pmatrix} \dot{\theta}_1 \\ \dot{\theta}_3 \end{pmatrix} = 0. \quad (2.25)$$

This linear system admits a non-null solution for  $\dot{\theta}_1$  and  $\dot{\theta}_3$  (i.e., a singular configuration occurs for the 2RS-2US structure) if and only if

$$m_{11}m_{22} - m_{12}m_{21} = 0. \quad (2.26)$$

The above relationship is the analytic expression of the singularity condition of the 2RS-2US structure. It is satisfied either when the two vectors  $\mathbf{m}_i = (m_{1i}, m_{2i})^T$ ,

$i = 1, 2$ , are parallel or when at least one of them is a null vector. The dimensionless parameters

$$k_1 = \frac{\|\mathbf{m}_1\|}{\|\mathbf{m}_2\|}, \quad k_2 = \frac{\|\mathbf{m}_1\| \|\mathbf{m}_2\|}{\|\mathbf{m}_1 \cdot \mathbf{m}_2\|}, \quad (2.27)$$

can be used to evaluate how far from a singularity a configuration is. At this point it, is reasonable to take as the farthest-from-singularity configuration the one where  $k_1$  is equal to 1 and  $k_2$  is equal to infinity; whereas a singular configuration occurs when at least one of the following conditions occur: (a)  $k_1 = 0$ , (b)  $k_1 = \infty$ , and (c)  $k_2 = 1$ . Based on these values, the following objective function, to be maximized during the robot motion, can be defined

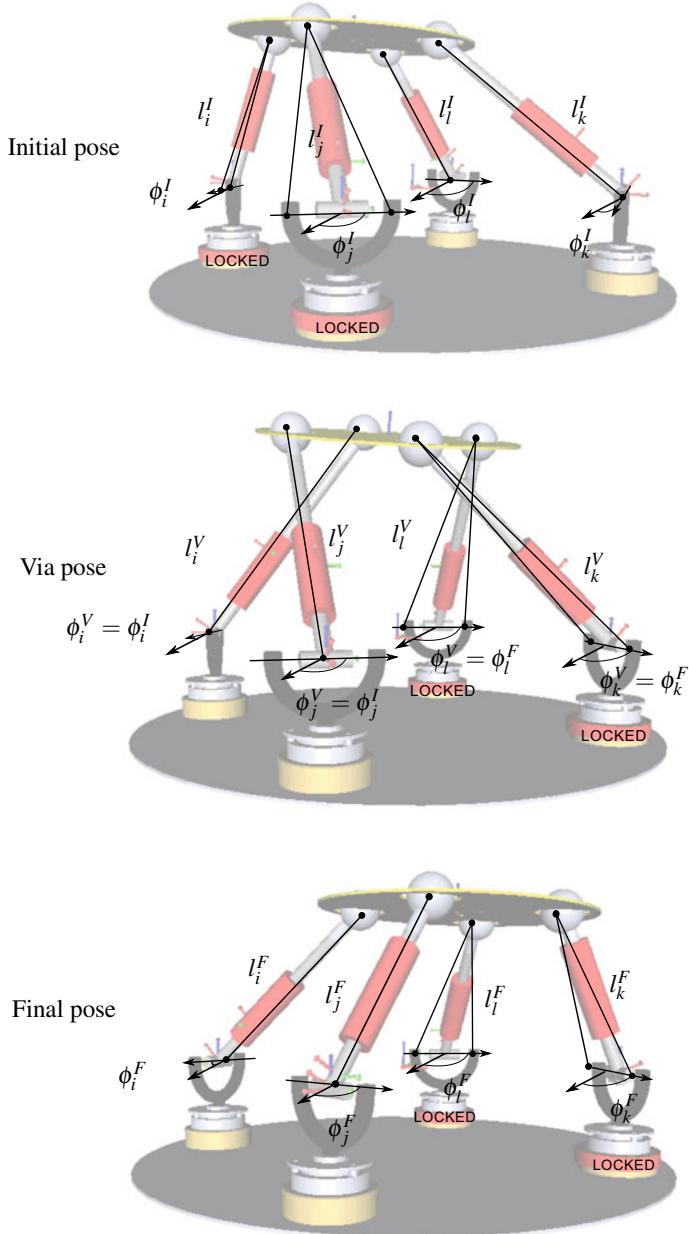
$$c = \frac{k_1}{(k_1 - 1)^4} + (k_2 - 1). \quad (2.28)$$

The value of such a function tends to infinity when  $k_1$  ( $k_2$ ) tends to 1 (infinity); and it decreases when either  $k_1$  ( $k_2$ ) tends to 0 (1) or  $k_1$  tends to infinity. Although it is well-known that a distance to a singularity, in a configuration space where translations and rotations coexist, cannot be defined, the above cost function will be useful later, as a heuristics, when assigning a cost to a path.

## 2.2 Maneuvers

Let us assume that we want to generate a trajectory connecting  $\mathbf{X}_I = (\mathbf{l}_I, \phi_I) = (l_1^I, \dots, l_4^I, \phi_1^I, \dots, \phi_4^I)$  to  $\mathbf{X}_F = (\mathbf{l}_F, \phi_F) = (l_1^F, \dots, l_4^F, \phi_1^F, \dots, \phi_4^F)$  where  $l_i$  is the length of leg  $i$  and  $\phi_i$  is the angle formed by  $\hat{\mathbf{w}}_{li}$  and the  $x$ -axis of the world reference frame (see Fig. 2.1). Since the robot is not able, in general, of reaching the final pose directly, it is necessary to introduce an intermediate one. This intermediate pose will be called *via pose* and it will be denoted by  $\mathbf{X}_V = (\mathbf{l}_V, \phi_V)$ . At this via pose the lockable joints are switched. The leg lengths at the via pose,  $\mathbf{l}_V$ , can be computed numerically by setting the released joints to their values in the final pose and solving a local optimization problem starting from the initial pose. This can be efficiently implemented using the Newton's method [15]. Then, the proposed maneuver consist of the two steps represented in Fig. 2.3.

Note that there are up to six sets of possible maneuvers connecting two given poses: one for each possible pair of locked joints. Once we have a candidate for a maneuver, and its corresponding via pose, it must be executed by driving the robot's prismatic actuators, as explained above. The simplest approach consists in linearly interpolating the leg lengths from  $\mathbf{l}_I$  to  $\mathbf{l}_V$ , and then from  $\mathbf{l}_V$  to  $\mathbf{l}_F$ . During this process, it might happen that the system reaches a different solution from the expected one (remind that the forward kinematics problem has no single solution). This might happen mainly when the maneuver involves a path close to a singularity. For the sake of simplicity, the obtained maneuver will be simply discarded.



**Fig. 2.3** The proposed maneuver connecting two configurations,  $X_I = (\mathbf{l}_I, \phi_I)$  and  $X_F = (\mathbf{l}_F, \phi_F)$  passing thought the via pose  $X_V = (\mathbf{l}_V, \phi_V)$ , with  $\phi_V = (\phi_i^I, \phi_j^I, \phi_k^F, \phi_l^F)$  and  $\mathbf{l}_V$  to be found, is performed in three steps. Top: with clutches connected to legs  $i$  and  $j$  engaged, and with clutches connected to legs  $k$  and  $l$  disengaged, the prismatic actuators are driven from  $\mathbf{l}_I$  to  $\mathbf{l}_V$ ; Center: the clutches states are inverted so that now clutches  $k$  and  $l$  are engaged and clutches  $i$  and  $j$  disengaged; Bottom: the prismatic actuators are driven from  $\mathbf{l}_V$  to  $\mathbf{l}_F$

There are more reasons to discard a maneuver: those giving rise to collisions or those in which not all joints remain within their valid range of motion. A complete test for collision detection and interference detection can be implemented using available collision detection packages such as GJK, SOLID, V-Clip, I-Collide, etc. (see [1, p. 201] and the references therein).

Once all valid maneuvers are computed, it is reasonable to choose the one that keeps the robot as far as possible from singularities. Unfortunately, there is no proper distance to a singularity [13]. As a simplification in our particular design, the quality measure to decide whether the maneuver is close to a singularity is taken to be (2.28) evaluated in the corresponding via pose. It is assumed that the bigger this value is, the farther the via pose is from a singularity. Then, the reciprocal of this value is taken as the cost of a maneuver.

The above procedure to find the best maneuver connecting two arbitrary configurations is summarized in pseudocode in Algorithm 1. Function CANDIDATE implements the Newton's method that computes the leg lengths in the via pose. Function VALID-PATH verifies if the final configuration is reached by linearly interpolating the leg lengths, checks if no collisions arise, and verifies if the joints are kept within their range along the trajectory. Finally, function COST assigns a cost to the maneuver based on the objective function (2.28) to a singularity of the via pose.

It is clear that the above algorithm might fail to find a path mainly when the initial and final poses are far apart in the configuration space of the robot. In these cases, one alternative is to subdivide the trajectory into segments whose initial and final poses can be connected using the above algorithm. Unfortunately, this simple idea might also fail. The alternative is to use a motion planner, as described in the next section.

## 2.3 Motion Planning

There are many possible approaches for implementing a motion planner but those based on probabilistic road maps [14] have demonstrated their potential in many applications [1, Chap. 7]. This approach has already been successfully applied to ordinary parallel robots in [3]. Next, it is adapted to the proposed robot. Within this approach, the robot would be subjected to a *learning phase* where its configuration space is randomly sampled. These samples are connected to their neighbors through the maneuvers, presented in the previous section, to generate a road map. Then, in the *query phase*, in which a path between two arbitrary poses must be found, the initial and final poses are firstly linked to their neighbors in the road map and, using a graph search algorithm, a path is found.

**Algorithm 1** BESTVIAPOSE( $X_i, X_j$ )

---

```

1: Maneuvers  $\leftarrow \{[1,2,3,4], [1,3,2,4], [1,4,2,3], [2,3,1,4], [2,4,1,3], [3,4,1,2]\}$   $\triangleright$  The first two indices
   of each 4-tuple correspond to the locked joints during the first motion of the maneuver
2:  $X_V \leftarrow \text{void}$ 
3: for all  $M \in \text{Maneuvers}$  do
4:    $[i, j, k, l] \leftarrow M$ 
5:    $\phi_x[i] \leftarrow \phi_I[i]$ 
6:    $\phi_x[j] \leftarrow \phi_I[j]$ 
7:    $\phi_x[k] \leftarrow \phi_F[k]$ 
8:    $\phi_x[l] \leftarrow \phi_F[l]$ 
9:    $I_x \leftarrow \text{CANDIDATE}(X_I, \phi_x, M)$ 
10:   $X_x \leftarrow (I_x, \phi_x)$ 
11:  if VALIDPATH( $X_I, X_x, X_F$ ) == TRUE then
12:     $\triangleright$  The maneuver is valid
13:    if COST( $X_x$ ) < COST( $X_V$ ) then
14:       $\triangleright$  COST(void) returns  $\infty$ 
15:       $X_V \leftarrow X_x$ 
16:    end if
17:  end if
18: end for
19: return  $X_V$ 
```

---

**2.3.1 Generating the Road Map**

The road map is built by sampling poses in the configuration space of the robot. When a sample is chosen, the best maneuvers to connect it to its neighboring poses previously generated are computed (see Fig. 2.4). Two poses are considered to be neighbors if the Euclidean norm between both their position and orientation components are below a given threshold. If a valid maneuver is found, its corresponding via pose is stored in an adjacent matrix together with its associated cost. If not, the stored cost will be infinite. Algorithm 2 gives this description in pseudocode.

To increase the density of the road map, it is always possible to add an intermediate configuration when two configurations fail to be connected directly through one of the six maneuvers that can be obtained using the described procedure.

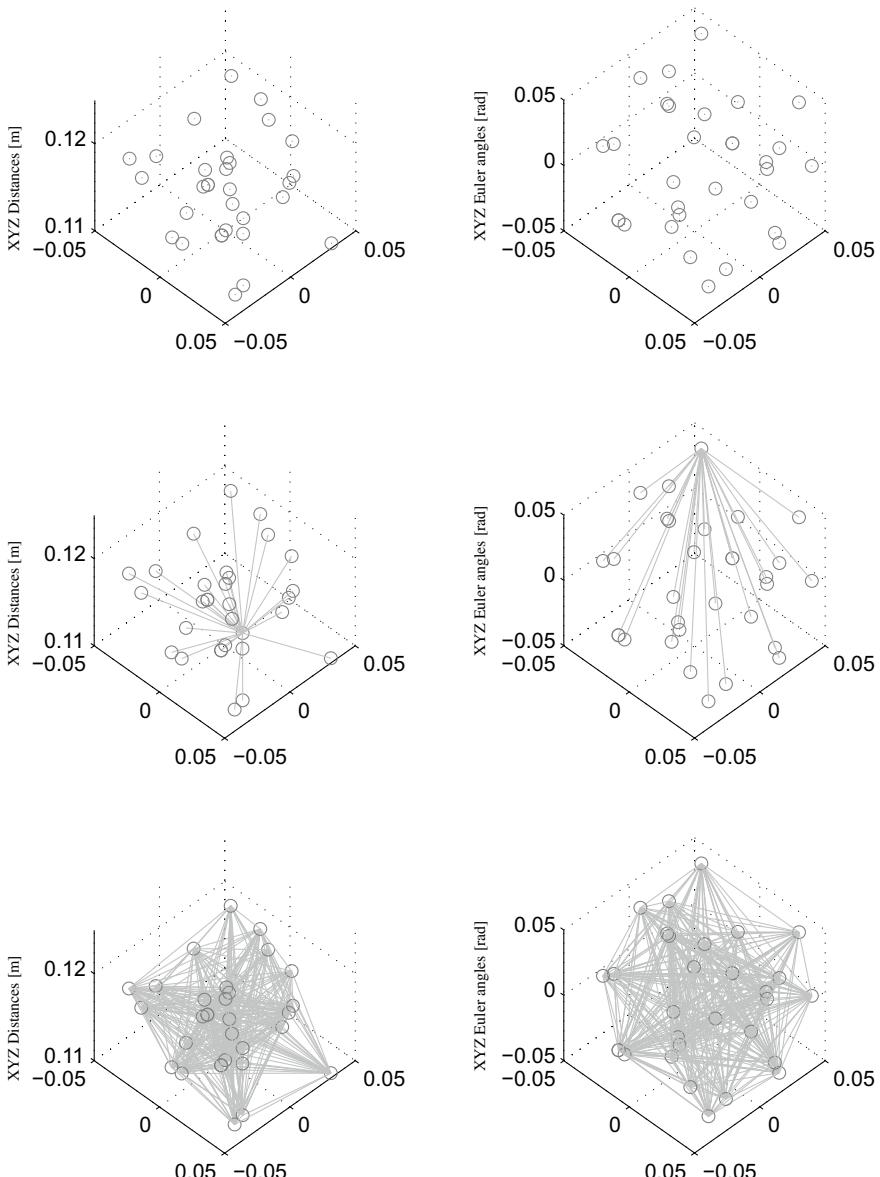
**Algorithm 2** GENERATEROADMAP

---

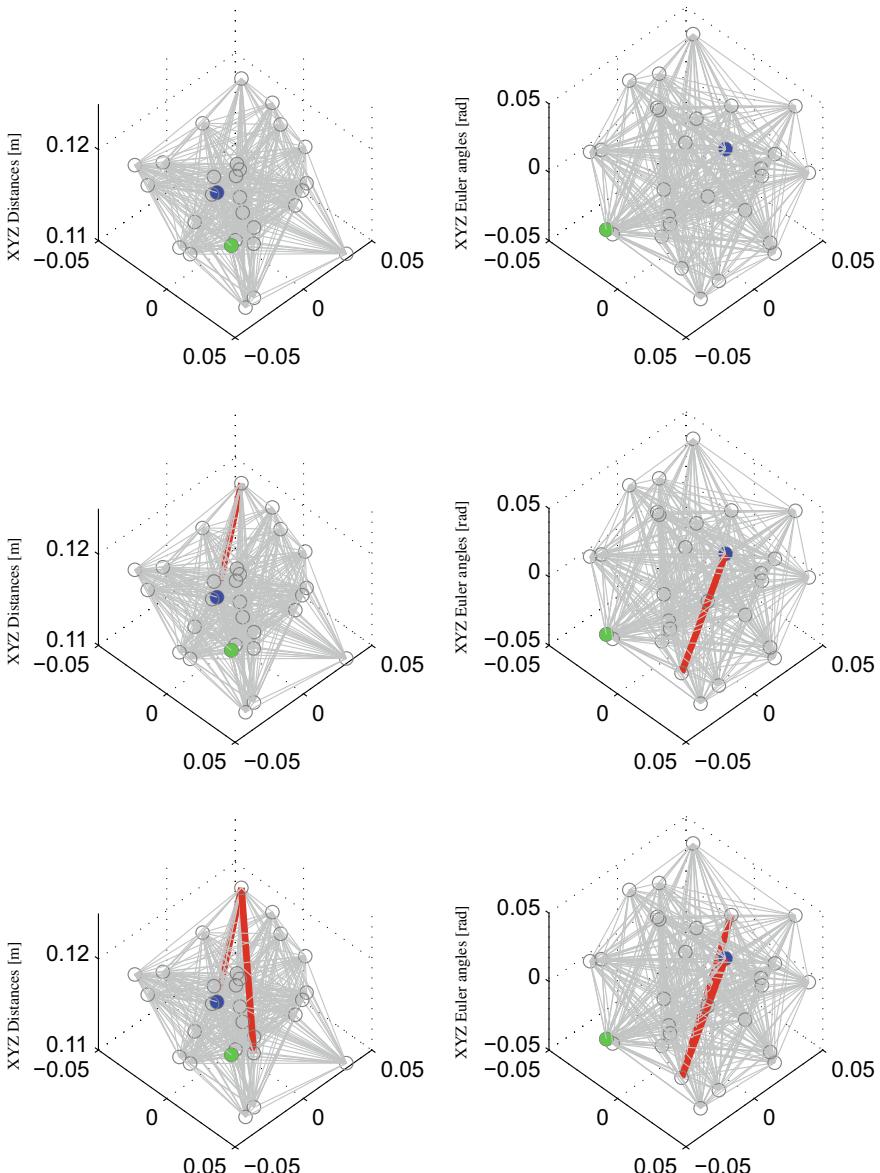
```

1: for  $i = 1$  to NumPoses do
2:    $X_i \leftarrow \text{RANDOMPOSE}()$ 
3:   Poses  $\leftarrow \text{FINDNEIGHBORPOSES}(X_i)$ 
4:   for all  $X_j \in \text{Poses}$  do
5:      $X_V \leftarrow \text{BESTVIAPOSE}(X_i, X_j)$ 
6:     ManMatrix[i,j]  $\leftarrow X_V$ 
7:     CostMatrix[i,j]  $\leftarrow \text{COST}(X_V)$ 
8:      $\triangleright$  COST(void) returns  $\infty$ 
9:   end for
10: end for
```

---



**Fig. 2.4** Learning phase of the proposed motion planning algorithm. As a result, a road map is generated. Top row: Random generation of poses. Center row: Connection generation between neighboring poses. Bottom row: Resulting road map. The left and right columns correspond to the subspaces of rotations and translations, respectively



**Fig. 2.5** Query phase of the proposed motion planning algorithm. The initial and final poses are represented in blue and green, respectively. Top row: The initial and final poses are connected to the road map. Center row: The starting pose is connected to the via pose. Bottom row: The via pose is connected to the end pose. The left and right columns correspond to the subspaces of rotations and translations, respectively

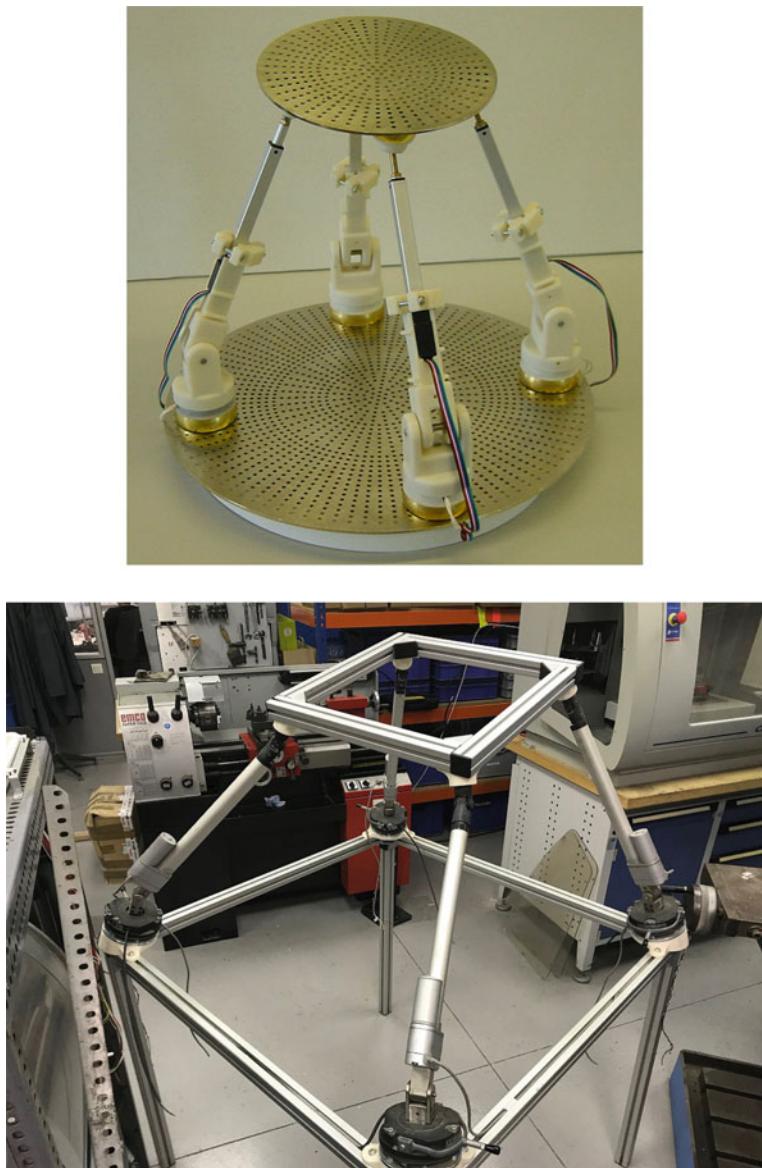
### 2.3.2 Finding a Path

If a trajectory—free from collisions and as far as possible from any singularity—connecting  $X_I$  to  $X_F$  must be generated, it is first necessary to connect these two poses to the previously generated road map (see Fig. 2.5). That is, the best maneuvers to connect them to their neighbors should be computed. Once the initial and final poses are connected to the road map, we need to find the shortest path connecting them in terms of cost. Dijkstra's algorithm is well-suited to this end [2, p. 595]. Finally, when the path is obtained, if one exists, the corresponding maneuvers—described in terms of leg lengths settings and sequences of locked and released revolute joints—can be executed by the robot.

## 2.4 Hardware Implementation

After verifying the kinematic behavior of the proposed robot in simulation, the prototype in Fig. 2.6 (top) was built. The base and the moving platform were made of 3 mm thick nickel-plated steel plates. They are disks of 400 mm and 200 mm in diameter, respectively. The lockable revolute joints were implemented using electromagnetic clutches manufactured by Huco-Dynatork Co. [9]. When one of these clutches is energized, the joint is free to rotate, otherwise it remains locked. The actuated prismatic joints were implemented using miniature servo linear motors manufactured by Firgelli Technologies Inc. [5]. The four actuators were controlled through a USB servo card. An interesting feature of this prototype is that the legs are attached to the base and the moving platform through magnetic fixtures. This simplifies any leg rearrangement during tests. Finally, all plastic elements were manufactured using a 3D printer.

It was observed, even when all clutches were engaged, that the moving platform was slightly shaky because the lockable joints were not completely stiff. This was directly connected with the diameter of the clutch and the mechanical play of the linear actuators and the clutches. As a consequence, the second prototype shown in Fig. 2.6 (bottom) was implemented, using more robust elements. It was implemented using EMTorq industrial clutches [4] and HDLS-12.00-2.00 linear servos from ServoCity [19]. While in the first version, one of the main goals was to have a robot where the  $R_b$  joints could be easily relocated, in the second version, the relocation objective was left aside. Both, moving platform and base, are squared-shaped built from metallic Item profiles [12]. A video of this implementation can be found in [6]. This video shows how the robot changes their leg lengths from their maximum to their minimum and the other way round. This is performed using several maneuvers in which the  $R_b$  joints of legs 2 and 4 are locked, while those of legs 1 and 3 are unlocked, and vice versa. As a result, the moving platform rotates about the  $z$ -axis.



**Fig. 2.6** The two implemented prototypes

## 2.5 Software Implementation

In order to verify the behavior of the proposed parallel robot executing the sequence of actions generated by the described path planner, a simulator using MATLAB and Simulink, whose output is connected to a VMRL 3D model of the robot, was implemented. A typical output of this simulator can be seen by following the link in [7].

As a simulation example, let us consider the case in which the diameters of the base and the moving platform are 0.4 m and 0.2 m, respectively. When the legs are extended at half their maximum extension, the moving platform is located at 0.3 m from the base. This is taken as the home configuration. The generated road map has been obtained by taking 100 configurations randomly sampled in a region centered at the home configuration with  $x \in [-0.04, 0.04]$ ,  $y \in [-0.04, 0.04]$ ,  $z \in [0.115, 0.125]$ ,  $\theta_x \in [-0.05, 0.05]$ ,  $\theta_y \in [-0.05, 0.05]$ , and  $\theta_z \in [-0.05, 0.05]$  (where distances are given in meters and the orientation angles in radians using the roll-pitch-yaw convention). When each of these configurations have been tried to be connected to all others, 2,275 connections fail (out of the 4,950 possible connections) for the six possible maneuvers. If an intermediate configuration is added in these cases, the amount of failed connections drops to 644. Due to these intermediate configurations, the total number of configurations in the road map is 3,229 and the total number of maneuvers checked for validity amounts to 114,420. 92,245 are discarded for different reasons (e.g., they force the prismatic actuators going out of their range, or force the U or the S joints extending over the limit of their maximum angles, etc.). Table 2.2 compiles this information.

**Table 2.2** Statistics for the road map generated in the example

Number of random configurations	100
Intermediate configurations added	3229
Connections	
Evaluated connections	19070
Possible direct connections	4950
Failed direct connections	2275
Failed after adding one intermediate configuration	644
Established connections	9356
Maneuvers	
Evaluated (6 per connection)	114420
Discarded	92245
Go outside joint limits	64180
Do not converge to a solution	87967
Lead to collisions	201

It is worth finally noting that, following the same reasoning that has lead us to derive the  $4R_bRPS$  parallel robot from the Gough–Stewart platform, it would be possible to replace the  $UPS$  legs of a spherical S-3 $UPS$  parallel robot with  $R_bRPS$  legs to derive a reconfigurable S-2 $R_bRPS$  spherical robot whose moving platform can be moved in a three-dimensional configuration space by using only two actuators and by introducing via poses, configurations where the locked  $R_b$  pair is changed. The analysis of such a robot would be similar to the one presented in this chapter.

## References

1. H. Choset, K.M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L.E. Kavraki, S. Thrun, *Principles of Robot Motion: Theory, Algorithms, and Implementations*, Intelligent Robotics and Autonomous Agents Series (2005)
2. T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, *Introduction to Algorithms* (2001). ISSN 1084-6654
3. J. Cortés, T. Siméon, Probabilistic motion planning for parallel mechanisms, in *IEEE International Conference on Robotics and Automation*, Taipei, Taiwan (2003), pp. 4354–4359
4. EMTorq, Industrial Electric Clutches and Brakes (2015). <http://www.emtorq.com/>
5. Firgelli Technologies Inc., Innovator in the Micro-Motion (2015). <http://www.firgelli.com/>
6. P. Grosch, 4bRRPS Robot Pose Rotation Around z-axis (video) (2015). <https://youtu.be/6MSQhsyw6uM>
7. P. Grosch, 4bRRPS Robot Simulator (video) (2015). <http://youtu.be/EXSNdvIOI9Y>
8. P. Grosch, R. Di Gregorio, J. Lopez, F. Thomas, Motion planning for a novel reconfigurable parallel manipulator with lockable revolute articulations, in *Proceedings of the IEEE International Conference on Robotics and Automation*, Anchorage, AK, USA (2010), pp. 4697–4702
9. Huco-Dynatork Co., World Leading Manufacturers of Small Precision Couplings (2015). [www.huco.com/](http://www.huco.com/)
10. C. Innocenti, Direct kinematics in analytical form of the 6–4 fully-parallel mechanism. ASME J. Mech. Des. **117**(1), 89–95 (1995)
11. C. Innocenti, V. Parenti-Castelli, Direct position analysis of the stewart platform mechanism. Mech. Mach. Theory **25**(6), 611–621 (1990)
12. Item Co., Building Kit Systems for Industrial Applications (2015). <http://www.item24.co.uk/>
13. J.P. Merlet, Jacobian, manipulability, condition number, and accuracy of parallel robots. J. Mech. Des. **128**(1), 199–206 (2006)
14. L.E. Kavraki, P. Svestka, J.C. Latombe, M.H. Overmars, Probabilistic roadmaps for path planning in high-dimensional configuration spaces. IEEE Trans. Robot. Autom. **12**(4), 566–580 (1996)
15. C.T. Kelley, *Solving Nonlinear Equations With Newton's Method*, Fundamentals of Algorithms (2003)
16. Y. Li, Y. Song, Z. Feng, C. Zhang, Complete Jacobian matrix of a class of incompletely symmetrical parallel mechanisms with 4-DoF. Chin. J. Mech. Eng. **43**(6), 37–40 (2007)
17. Y. Lu, J. Xu, Simulation solving/modifying velocity and acceleration of a 4UPS+SPR type parallel machine tool during normal machining of a 3D free-form surface. Int. J. Adv. Manuf. Technol. **42**(7–8), 804–812 (2009)
18. Y. Lu, M. Zhang, Y. Shi, J. Yu, Kinematics and statics analysis of a novel 4-DoF 2SPS+2SPR parallel manipulator and solving its workspace. Robotica **27**(5), 771–778 (2009)
19. Servo City Co., Mechanical Components for use in Robotics (2015). <http://www.servocity.com/>
20. A. Sokolov, P. Xirouchakis, Kinematics of a 3-DoF parallel manipulator with an R-P-S joint structure. Robotica **23**(2), 207–217 (2005)

# Chapter 3

## Spherical Non-holonomic Joints



In this chapter, a method to generate under-actuated robots by substituting ordinary spherical joints (S joints) with non-holonomic spherical joints is described, a topic that was already treated in [3, 4]. The practicalities of implementing non-holonomic spherical joints are also included.

As already mentioned in the introduction, non-holonomic constraints arise in many different areas of robotics such as motion planning and control of mobile robots, reorientation of free-flying space robots, rolling contacts of multi-fingered hands, etc. In all these cases, the non-holonomic constraints are inherent to the problem, but there are some cases in which the artificial introduction of this kind of constraints can provide important advantages.

At this point, we can distinguish between the number of degrees of freedom of the configuration space, also called configuration (or finite) degrees of freedom [1], which is defined as the minimum number of geometric parameters necessary to uniquely identify the configuration of the mechanical system [6], and the instantaneous degrees of freedom, also called velocity degrees of freedom [1], of the same mechanical system.

As presented in the introduction of this book, the literature on the use of non-holonomic devices in the design of robots is limited to few examples. Despite the difference of purpose, all mentioned examples include a surface (cylinder, sphere or plane) in contact with a roller that can freely roll without slipping laterally. This no-slip constraint is a non-holonomic constraint, a constraint relating the velocities of the surface and the roller. For the sphere-roller case, the kinematics of the assembly is equivalent to that of a unicycle on a sphere whose equations of motion can be represented by first-order differential equations [5].

Many efforts have been made to clarify different aspects of non-holonomic mechanical systems including its controllability, stability, feedback stabilization, time-periodic control, chained form transformation, etc. but, in any case, achieving a formulation for the kinematics of the system, as compact and simple as possible, is

essential to explore the applicability of all these results available in the specialized literature.

In this chapter, the under-actuated parallel robot presented by Ben-Horin and Thomas in [2] is reconsidered from a different point of view. This permits seeing it as a particular case of under-actuated robot obtained through the substitution of spherical joints with non-holonomic spherical joints. This paves the way for the compact formulation of its kinetostatic presented in Chap. 4.

This chapter is simply structured in two sections: Sect. 3.1 describes how to generate under-actuated parallel robots from fully parallel robots, and Sect. 3.2 gives practical considerations on the implementation of non-holonomic spherical joints.

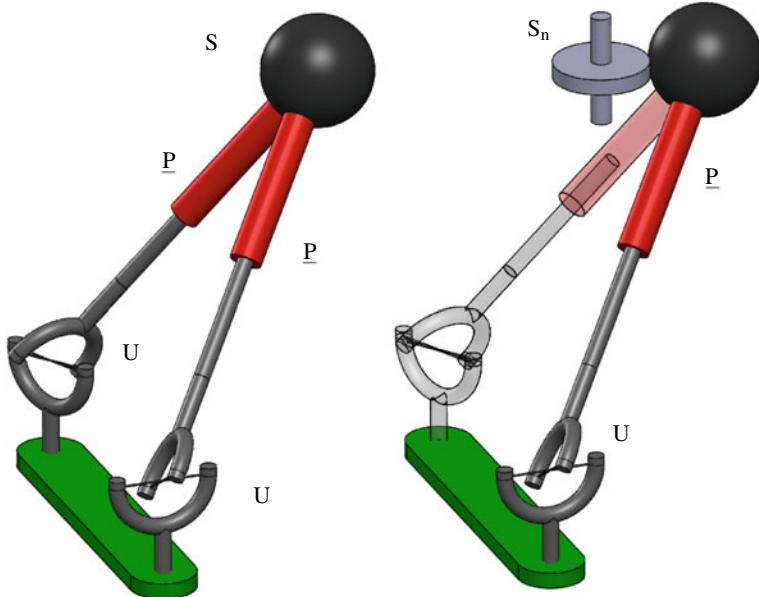
### 3.1 Under-Actuated Parallel Robots with Spherical Non-holonomic Joints

Two rigid bodies connected by a spherical joint can assume any relative orientation, and can move from one relative orientation to another by covering any spherical-motion path that joins the two relative orientations. Actually, the possibility of freely orientating two rigid bodies with respect to one another is not related to the possibility of performing relative rotations around axes which pass through the center of spherical motion and have any direction. In fact, a suitable sequence of finite rotations around coplanar axes that pass through the spherical-motion center can freely orientate one rigid body with respect to another. Thus, if the only free relative orientation of two rigid bodies is required, the use of a spherical joint will be redundant. The use of a joint that allows only rotations around coplanar axes that pass through a fixed point would be sufficient.

Due to frictional forces, the rolling contact between a sphere and a roller forbids the sphere rotations around the axis through the sphere center, and perpendicular to the plane defined by the roller axis and the sphere center. By combining such a non-holonomic constraint with other constraints that forbid the relative translation between the sphere center and the roller axis, a non-holonomic joint will result. This joint constrains two rigid bodies: one fixed to the sphere and the other fixed to the plane, defined by the roller axis and the sphere center. Then, the resulting constrained motion permits only relative rotations around axes lying on the above-mentioned plane and passing through the sphere center.

The constraint forces, which two rigid bodies, joined by an  $S_n$  joint, exert on one another through the joint, can be reduced to a resultant force applied on the sphere center and a torque perpendicular to the plane defined by the roller axis and the sphere center. This torque is the static effect of the non-holonomic constraint, whereas the resultant force on the sphere center is the same static effect that an  $S$  joint would have generated.

In general, two rollers, whose axes define with the sphere center two different planes, constrains the sphere to rotate about the intersection line between these two

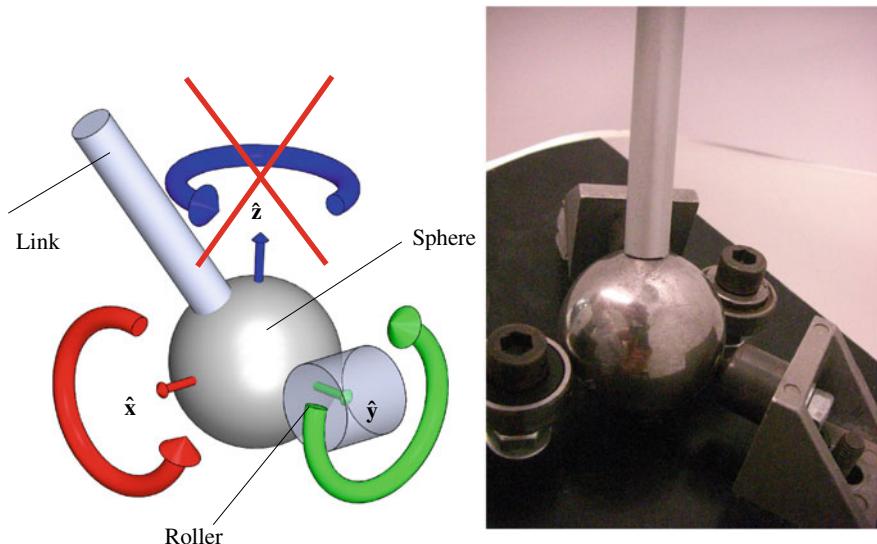


**Fig. 3.1** When substituting the multiple spherical joint shared by two UPS legs by a  $S_n$  joint, a  $UPS_n$  leg is obtained

planes. Three rollers whose axes define with the sphere center three different planes lock the sphere. Then, the maximum torque transmitted through a  $S_n$  joint, due to its frictional origin, can be fixed by suitably choosing the number of roller-sphere contacts together with the normal force transmitted through each contact.

The above discussion leads us to conclude that the substitution of a number of  $S$  joints by  $S_n$  joints in a kinematic chain does not change the configuration space of that chain. That is, the reachable configurations remain unaltered. It only affects the instantaneous kinematics of the mechanism by reducing the practicable paths for moving that chain from one configuration to another. Moreover, due to the torque that arises in a  $S_n$  joint in a parallel robot, the substitution of a number of  $S_n$  joints for as many  $S$  joints, see Fig. 3.1, allows the elimination of a number of actuators equal to the number of introduced  $S_n$  joints (i.e., the substitution leads to an under-actuated robot).

Due to the high number of  $S$  joints appearing in fully parallel robots, the substitutions of  $S_n$  joints for  $S$  joints, accompanied by as many eliminations of actuators in the prismatic joints, can be operated in many ways. By exploiting all the possible substitutions, new under-actuated parallel robots can be generated. It is worth noting that a passive UPS leg only affects the workspace boundaries due to the range of motion of the  $P$  joint. If this effect is not necessary, the elimination of the actuator in a prismatic joint could be accompanied by the elimination of the whole resulting passive UPS leg.



**Fig. 3.2** Conceptual design (left) and proof of concept (right) of the proposed non-holonomic joint

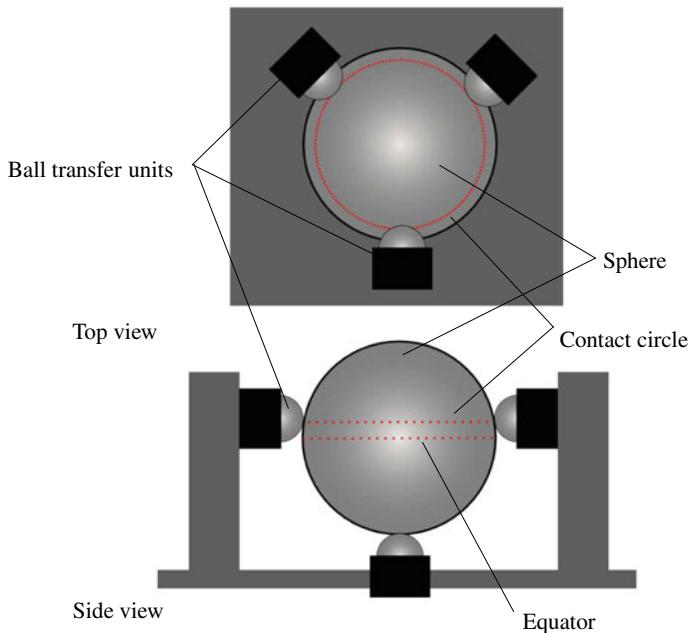
### 3.2 Implementation of Spherical Non-holonomic Joints

The considered non-holonomic spherical joint consists of a spherical joint in contact with a roller as shown in Fig. 3.2 (left). In a  $3S_nPU$  robot the non-holonomic joints link the fixed base to prismatic actuators, and in a  $S_n-2SPU$  spherical robot the non-holonomic joint directly links the base to the moving platform (see Chaps. 4 and 6, respectively).

To get some insight into the behavior of the proposed joint, the prove of concept shown in Fig. 3.2 (right) was implemented. Some experiments with this assembly led us to establish the following requirements for the final design:

1. The sphere should be secured to avoid it translating, but letting it to freely rotate.
2. The roller should be put in contact with the sphere with a high contact force. The coefficient of friction and the magnitude of the force will determine the maximum moment that the sphere can undergo before it slips with respect to the roller.
3. The location of the roller and the securing elements must be chosen so that the range of motion of the leg is maximized.

Our design is also limited by the characteristics of the spheres commercially available. The chosen sphere is a stainless steel ball of 50 mm in diameter (RS code 687-629) drilled using an electric discharge machine to fasten the link. The chosen securing elements are four ball transfer units (RS code 687-679) arranged as shown in Fig. 3.3. This arrangement disables sphere's translation while still allowing it to

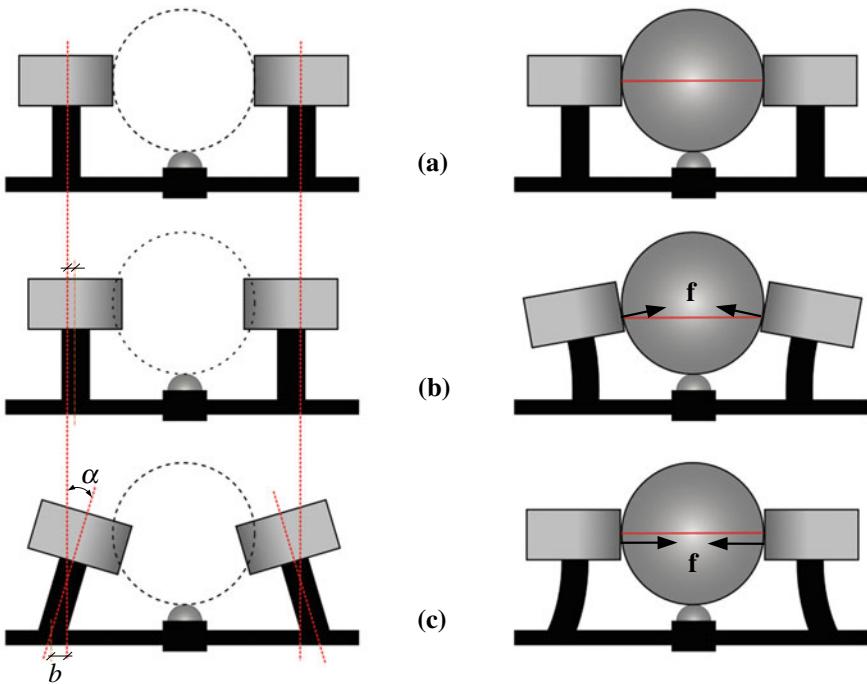


**Fig. 3.3** Securing the sphere within its housing

freely rotate. Observe how three contact points occur above the sphere's equator, thus preventing it from getting out of the housing.

To obtain the non-holonomic effect, we have to place a roller in contact with the sphere. One of the ball transfer units could actually be replaced with a roller. Nevertheless, this is not enough because a high contact force is needed to prevent slippage between the sphere and the roller. The high contact force on the sphere must be counteracted by the remaining ball transfer units, which is not possible as they can only support a low load while rotating freely. An alternative solution is to place two opposing rollers in contact with the sphere at polar points so that the counteraction on the ball transfer units is minimized. The introduction of the second roller also doubles the friction force thus doubling the allowed torque. Nevertheless, this option has an important drawback: the rollers' axes must be parallel and their contacts with the sphere must be exactly polar, which is not easy to achieve. If any of these two conditions is not satisfied, slippage at the contact points with the roller will occur, or the sphere will only rotate about a single axis. Therefore, the mechanism should include the possibility of adjusting the location of the rollers' axes to ensure the polarity of the contact points.

The required high contact forces between the rollers and the sphere can be obtained by taking advantage of the rollers' stems flexibility, as illustrated in Fig. 3.4. If the rollers are just touching the sphere, no forces are obviously generated (Fig. 3.4a). If the rollers are too close and parallel before placing the sphere, the rollers' axes will

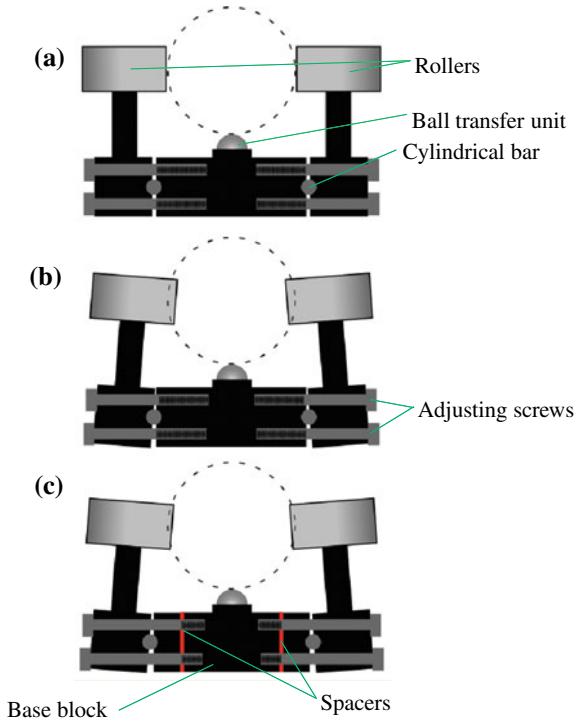


**Fig. 3.4** To guarantee a high contact force, a device is needed to adjust  $\alpha$  and  $b$

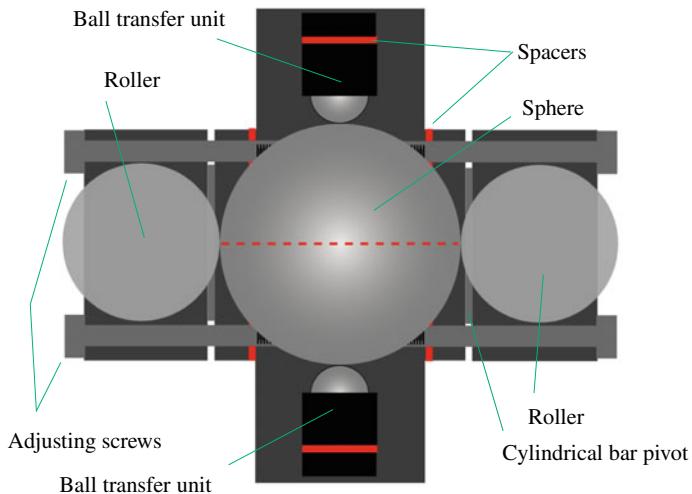
not be parallel and the contact points will not be polar (Fig. 3.4b). On the contrary, if the orientations and the anchor points of the rollers' stems can be adjusted by an angle given by  $\alpha$  and a translation given by  $b$  (Fig. 3.4c), the stems' rollers will bend when the sphere is placed so that the axes of the rollers become parallel and the contact points are polar. Nevertheless, since the values of  $\alpha$  and  $b$  are difficult to determine beforehand, a device to adjust them must be included in the design. Summing up, the design, besides including adjustments for the rollers axes orientations and the location of the contact points, should also include adjustments for  $\alpha$  and  $b$ .

Figure 3.5 shows the device designed to adjust  $\alpha$  and  $b$ . It consists of a central base block and two side blocks to which the rollers are attached. The side blocks are fasten to the central one by eight screws, four per side. Two cylindrical bars are placed between the blocks so that they can pivot on these bars and the angles between them can then be adjusted with the screws. The center block is split into three elements so that we have the possibility to add spacers between them. Adding or subtracting spacers changes the distance  $b$ . To adjust the lateral contact points between the rollers and the sphere, we have also added the possibility of adding spacers at the housing of the ball transfer units. This will move the sphere sideways as shown in Fig. 3.6.

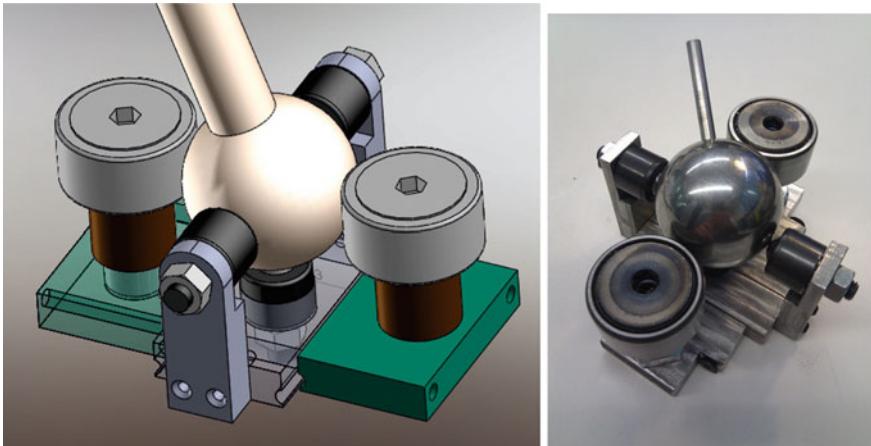
As a result of the above analysis, we came up with the CAD design shown in Fig. 3.7 (left). Its implementation can be seen in Fig. 3.7 (right).



**Fig. 3.5** Device to adjust the angles and the distance between the rollers' stems



**Fig. 3.6** Device to adjust the contact points between the rollers and the sphere



**Fig. 3.7** CAD design of the proposed non-holonomic joint (left), and its implementation (right)

The remaining of this book consists in the study of the kinetostatics and the motion planning problem for the spatial  $3S_nPU$  robot and the spherical  $S_n\text{-}2UPS$  robot. These robots result from the application of the described joint substitution to the  $6UPS$  spatial and the  $S\text{-}3PS$  spherical robot, respectively. Contrarily to what happens with parallel robots with lockable joints, it is worth including a separate analysis of the spherical case as it admits a simple analysis that cannot be extended to the spatial one.

## References

1. J. Angeles, *Fundamentals of Robotic Mechanical Systems: Theory, Methods, and Algorithms*, Mechanical Engineering Series (2003)
2. P. Ben-Horin, F. Thomas, A nonholonomic 3-DoF parallel robot, in *Advances in Robot Kinematics: Analysis and Design* (Springer, Berlin, 2008), pp. 111–118
3. P. Grosch, R. Di Gregorio, F. Thomas, Generation of under-actuated parallel robots with non-holonomic articulations and kinetostatic analysis of a case-study, in *Proceedings of the ASME International Design Engineering Technical Conference and Computers and Information in Engineering Conference*, San Diego, USA (2009), pp. 979–986
4. P. Grosch, R. Di Gregorio, F. Thomas, Generation of under-actuated manipulators with non-holonomic articulations from ordinary manipulators. *J. Mech. Robot.* **2**(1), 11005–11012 (2010)
5. M.P. Hennessey, Visualizing the motion of a unicycle on a sphere. *Int. J. Model. Simul.* **26**(1), 69–79 (2006)
6. O.M. O'Reilly, *Intermediate Dynamics for Engineers* (2008)

# Chapter 4

## Kinematics of the $3S_nPU$ Spatial Robot

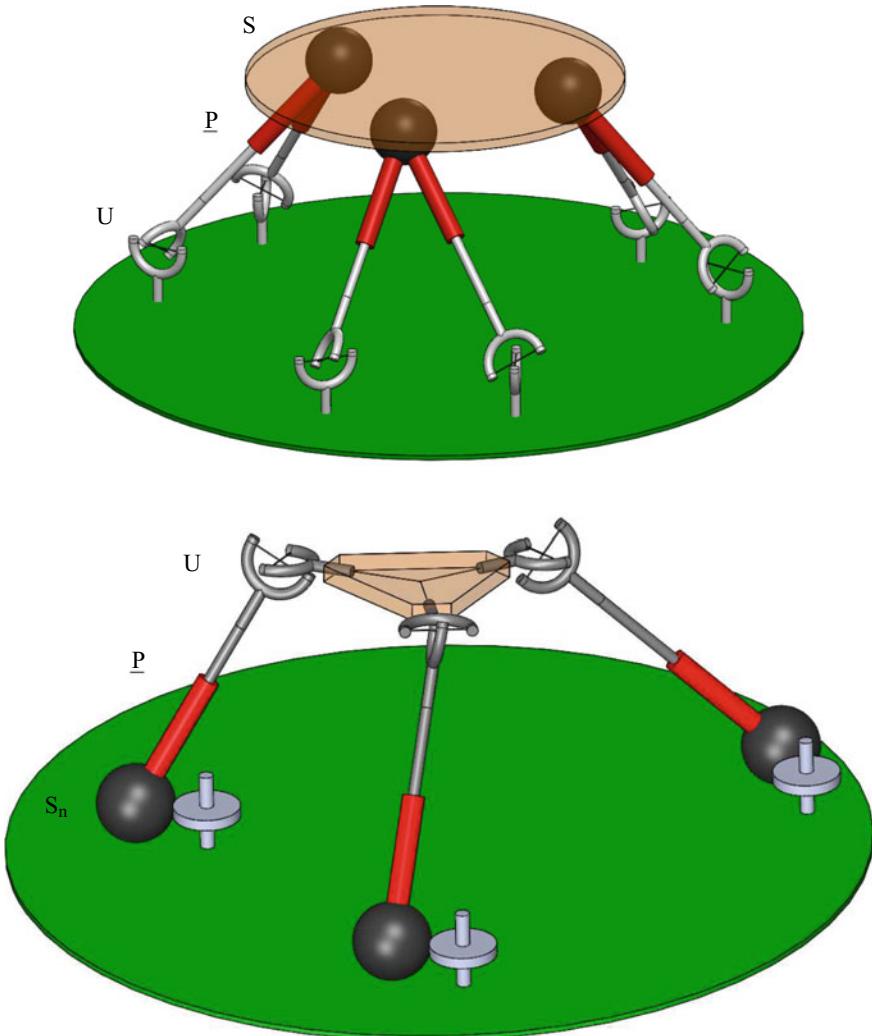


In this chapter, the under-actuated  $3S_nPU$  parallel robot presented in [1] is derived from the 6-3 fully-parallel robot. A compact formulation for its kinetostatics is presented, an essential result for its design and control. Part of the work presented in this chapter was carried out in collaboration with professor Raffaele di Gregorio, from the University of Ferrara, Italy. Part of this work appeared in [5, 6].

This chapter is structured as follows. Section 4.1 discusses how the  $3S_nPU$  robot can be derived from a 6-3 fully-parallel robot. Sections 4.2 and 4.3 are devoted to the analysis of its kinematics and statics, respectively, and how both are related through the computation of the instantaneous power balance. Sections 4.4 and 4.5 deal with its singularities and the related topic of its controllability, respectively. Finally, Sect. 4.6 presents an example where its singularities and controllability regions are numerically computed.

### 4.1 The $3S_nPU$ Robot

As explained in Chap. 1, a fully parallel robot with 6-3 architecture features three couples of UPS legs with three double S joints in the moving platform (Fig. 4.1 (top)). This architecture was first proposed by Stewart [13], in 1965, for a flight simulator. Later on, with the renewed interest for the parallel robots, started at the end of the eighties, it was profusely studied. In particular, regarding the direct position analysis of the 6-3 parallel robot, Innocenti and Parenti-Castelli [7] demonstrated that at most sixteen poses for the moving platform correspond to a given set of leg lengths. Then, Parenti-Castelli and Di Gregorio [10] demonstrated that the pose of the moving platform is uniquely determined when the value of one passive joint variable is measured besides the six leg lengths. The direct position analysis of this parallel robot can also be used for other spatial parallel robots that become 3RS structures when the actuators are locked (see, for instance, [3, 8, 11, 12]).



**Fig. 4.1** Top: a fully parallel robot with 6-3 architecture. Bottom: the under-actuated parallel robot with  $3S_nPU$  architecture resulting from applying the joint substitution presented in Chap. 3

Starting from the 6-3 architecture, each couple of UPS legs sharing an S joint (Fig. 4.1 (top)) can be transformed into an  $S_nPU$  leg, as explained in the previous chapter. By operating this substitution in all the three couples of UPS legs together with the inversion of the moving platform and the fixed base, the  $3S_nPU$  under-actuated robot, shown in Fig. 4.1 (bottom), is obtained.

Regarding the direct position analysis of the obtained robot, since its configuration space has six dimensions, a number of closure equations equal to the number of

unknowns can be written if, and only if, over the three legs lengths, three more passive joint variables are measured. Now, we can think of the three U joints as two revolute joints with intersecting axes. Then, if we measure the angle of the revolute joints adjacent to the prismatic joints, the closure equation system coincides with the one of the 6-3 parallel robot for the assigned leg lengths [10], and admits at most sixteen solutions for the pose of the moving platform. Moreover, if the joint variable of one revolute joint adjacent to the moving platform is also measured, only one pose for the moving platform will satisfy the closure equations [10].

The derivation of the input-output velocity relationships for a spatial fully parallel robot leads to satisfactory formulations of their Jacobian matrices. Nevertheless, this approach is not valid, in general, for lower-mobility parallel robots. In the case of the 3S<sub>n</sub>PU robot, this approach would lead to a  $3 \times 3$  Jacobian matrix whose analysis cannot predict all possible singularities [15]. A deeper analysis must be performed as detailed in Sect. 4.4.

## 4.2 Instantaneous Kinematics

Figure 4.2 shows the  $i$ th leg,  $i = 1, 2, 3$ , of a 3S<sub>n</sub>PU robot together with the notation that will be used.  $\hat{\mathbf{w}}_{1i}$  and  $\hat{\mathbf{w}}_{2i}$  are two mutually orthogonal unit vectors fixed to the base and lying on the plane defined by the roller axis and the center,  $A_i$ , of the sphere.  $\hat{\mathbf{w}}_{3i}$  and  $\hat{\mathbf{w}}_{4i}$  are the two mutually orthogonal unit vectors of the axes of the two revolute joints constituting the U joint.  $B_i$  is the center of the U joint.  $\mathbf{a}_i$  and  $\mathbf{b}_i$  are the two position vectors which locate points  $A_i$  and  $B_i$ , respectively, in a generic Cartesian reference fixed to the base, whereas  $\mathbf{p}$  is the position vector of a point in the moving platform,  $P$ , in the same reference frame.  $\theta_{ji}$ ,  $i, j = 1, \dots, 4$ , is a joint variable denoting a rotation angle around the axis defined by  $\hat{\mathbf{w}}_{ji}$ ,  $i, j = 1, \dots, 4$ , whose sign is given according to the right-hand rule with respect to  $\hat{\mathbf{w}}_{ji}$ . The length of the  $i$ th leg is equal to  $\|\mathbf{b}_i - \mathbf{a}_i\|$ , and it will be denoted  $l_i$ . Moreover, the leg-axis' unit vector,  $\hat{\mathbf{g}}_i$ , and the unit vector,  $\hat{\mathbf{h}}_i$  ( $\hat{\mathbf{r}}_i$ ), normal to the plane defined by the revolute axes of the U joint (by the roller axis together with the sphere center in the S<sub>n</sub>) satisfy the following relationships:

$$l_i \hat{\mathbf{g}}_i = \mathbf{b}_i - \mathbf{a}_i, \quad \hat{\mathbf{h}}_i = \hat{\mathbf{w}}_{3i} \times \hat{\mathbf{w}}_{4i}, \quad \text{and} \quad \hat{\mathbf{r}}_i = \hat{\mathbf{w}}_{1i} \times \hat{\mathbf{w}}_{2i}. \quad (4.1)$$

The time differentiation of the first relationship in (4.1) yields

$$\dot{l}_i \hat{\mathbf{g}}_i + l_i \dot{\hat{\mathbf{g}}}_i = \dot{\mathbf{b}}_i. \quad (4.2)$$

Since  $\dot{\hat{\mathbf{g}}}_i = (\dot{\theta}_{1i} \hat{\mathbf{w}}_{1i} + \dot{\theta}_{2i} \hat{\mathbf{w}}_{2i}) \times \hat{\mathbf{g}}_i$ , and  $\dot{\mathbf{b}}_i = \dot{\mathbf{p}} + \omega \times (\mathbf{b}_i - \mathbf{p})$ , where  $\omega$  denotes the angular velocity of the moving platform, Eq. (4.2) can be rewritten as:

$$\dot{l}_i \hat{\mathbf{g}}_i + l_i (\dot{\theta}_{1i} (\hat{\mathbf{w}}_{1i} \times \hat{\mathbf{g}}_i) + \dot{\theta}_{2i} (\hat{\mathbf{w}}_{2i} \times \hat{\mathbf{g}}_i)) = \dot{\mathbf{p}} + \omega \times (\mathbf{b}_i - \mathbf{p}). \quad (4.3)$$

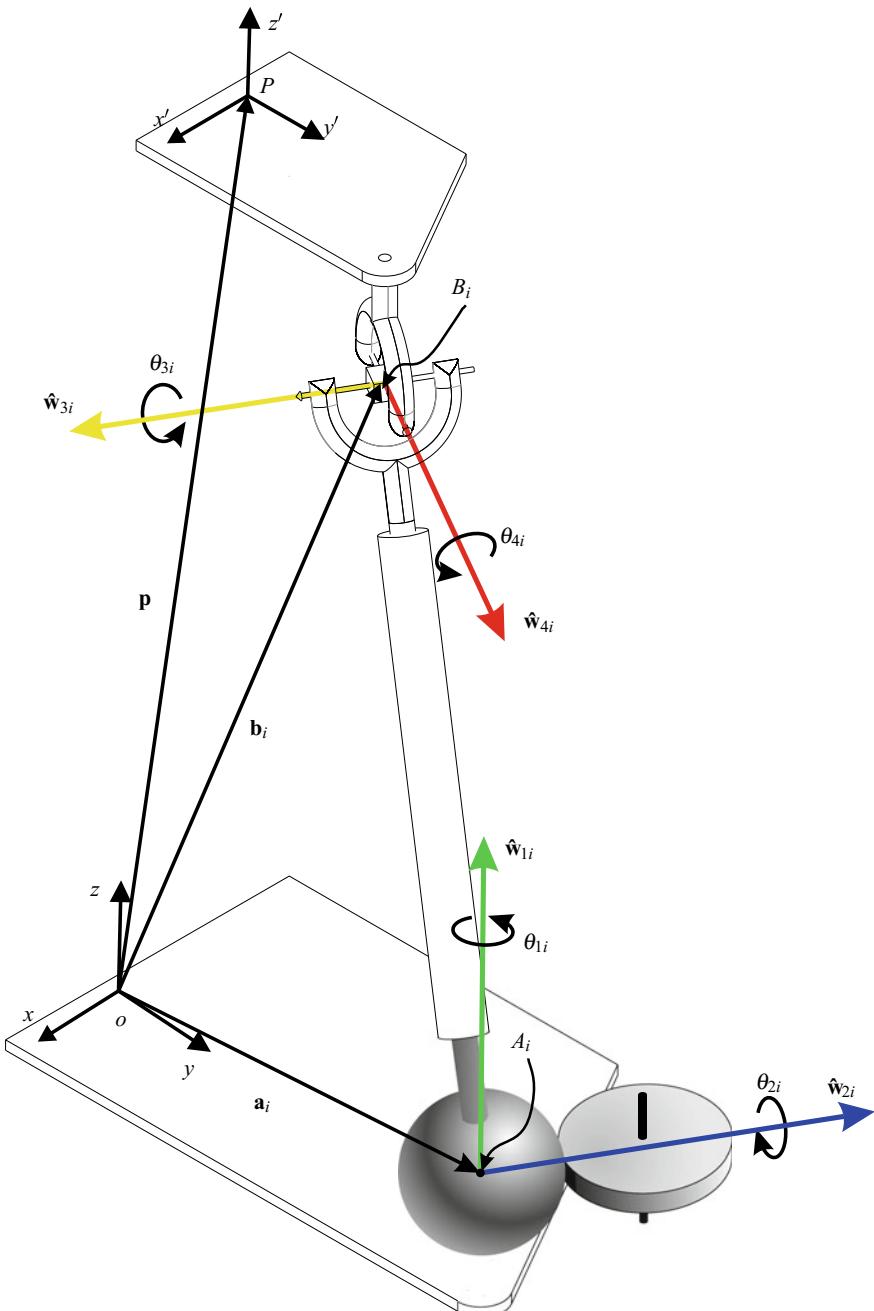


Fig. 4.2 Notation associated with the  $i$ th leg of the studied 3S<sub>n</sub>PU robot

The dot products of (4.3) by  $\hat{\mathbf{w}}_{1i}$  and  $\hat{\mathbf{w}}_{2i}$  yield the following two scalar equations:

$$\dot{l}_i (\hat{\mathbf{g}}_i \cdot \hat{\mathbf{w}}_{1i}) + l_i \dot{\theta}_{2i} ((\hat{\mathbf{w}}_{2i} \times \hat{\mathbf{g}}_i) \cdot \hat{\mathbf{w}}_{1i}) = \dot{\mathbf{p}} \cdot \hat{\mathbf{w}}_{1i} + \omega \times (\mathbf{b}_i - \mathbf{p}) \cdot \hat{\mathbf{w}}_{1i}, \quad (4.4)$$

$$\dot{l}_i (\hat{\mathbf{g}}_i \cdot \hat{\mathbf{w}}_{2i}) + l_i \dot{\theta}_{1i} ((\hat{\mathbf{w}}_{1i} \times \hat{\mathbf{g}}_i) \cdot \hat{\mathbf{w}}_{2i}) = \dot{\mathbf{p}} \cdot \hat{\mathbf{w}}_{2i} + \omega \times (\mathbf{b}_i - \mathbf{p}) \cdot \hat{\mathbf{w}}_{2i}. \quad (4.5)$$

On the other hand (see Fig. 4.2), the angular velocity of the moving platform is equal to  $\sum_{j=1,4} \dot{\theta}_{ji} \hat{\mathbf{h}}_j$ , whose dot product by  $\hat{\mathbf{h}}_i$  gives the following expression

$$\omega \cdot \hat{\mathbf{h}}_i = \dot{\theta}_{1i} (\hat{\mathbf{w}}_{1i} \cdot \hat{\mathbf{h}}_i) + \dot{\theta}_{2i} (\hat{\mathbf{w}}_{2i} \cdot \hat{\mathbf{h}}_i).$$

Solving (4.4) and (4.5) for  $\dot{\theta}_{2i}$  and  $\dot{\theta}_{1i}$ , respectively, and replacing the result in the above equation, yields

$$\begin{aligned} \omega \cdot \hat{\mathbf{h}}_i &= \left( \frac{\dot{\mathbf{p}} \cdot \hat{\mathbf{w}}_{2i} + \omega \times (\mathbf{b}_i - \mathbf{p}) \cdot \hat{\mathbf{w}}_{2i} - \dot{l}_i \hat{\mathbf{g}}_i \cdot \hat{\mathbf{w}}_{2i}}{l_i (\hat{\mathbf{w}}_{2i} \times \hat{\mathbf{g}}_i \cdot \hat{\mathbf{w}}_{2i})} \right) (\hat{\mathbf{w}}_{1i} \cdot \hat{\mathbf{h}}_i) \\ &\quad + \left( \frac{\dot{\mathbf{p}} \cdot \hat{\mathbf{w}}_{1i} + \omega \times (\mathbf{b}_i - \mathbf{p}) \cdot \hat{\mathbf{w}}_{1i} - \dot{l}_i \hat{\mathbf{g}}_i \cdot \hat{\mathbf{w}}_{1i}}{l_i (\hat{\mathbf{w}}_{1i} \times \hat{\mathbf{g}}_i \cdot \hat{\mathbf{w}}_{1i})} \right) (\hat{\mathbf{w}}_{2i} \cdot \hat{\mathbf{h}}_i). \end{aligned} \quad (4.6)$$

Taking into account the vector identities

$$\begin{aligned} \hat{\mathbf{h}}_i \times \hat{\mathbf{r}}_i &= \hat{\mathbf{h}}_i \times (\hat{\mathbf{w}}_{1i} \times \hat{\mathbf{w}}_{2i}) = (\hat{\mathbf{w}}_{2i} \cdot \hat{\mathbf{h}}_i) \hat{\mathbf{w}}_{1i} - (\hat{\mathbf{w}}_{1i} \cdot \hat{\mathbf{h}}_i) \hat{\mathbf{w}}_{2i}, \\ \hat{\mathbf{r}}_i \times \hat{\mathbf{g}}_i &= (\hat{\mathbf{w}}_{1i} \times \hat{\mathbf{w}}_{2i}) \cdot \hat{\mathbf{g}}_i = -(\hat{\mathbf{w}}_{1i} \times \hat{\mathbf{g}}_i) \cdot \hat{\mathbf{w}}_{2i} = (\hat{\mathbf{w}}_{2i} \times \hat{\mathbf{g}}_i) \cdot \hat{\mathbf{w}}_{1i}, \end{aligned}$$

the relationship (4.6) can be rewritten as

$$\dot{l}_i \hat{\mathbf{g}}_i \cdot (\hat{\mathbf{h}}_i \times \hat{\mathbf{r}}_i) = \dot{\mathbf{p}} \cdot (\hat{\mathbf{h}}_i \times \hat{\mathbf{r}}_i) + \omega \cdot ((\mathbf{b}_i - \mathbf{p}) \times (\hat{\mathbf{h}}_i \times \hat{\mathbf{r}}_i) - l_i (\hat{\mathbf{r}}_i \cdot \hat{\mathbf{g}}_i) \hat{\mathbf{h}}_i). \quad (4.7)$$

Since  $\dot{l}_i$  can also be obtained as the projection of  $\dot{\mathbf{b}}_i$  on  $\hat{\mathbf{g}}_i$  (see Eq.(4.2)), the following expression holds

$$\dot{l}_i = \dot{\mathbf{b}}_i \cdot \hat{\mathbf{g}}_i = \dot{\mathbf{p}} \cdot \hat{\mathbf{g}}_i + \omega \cdot [(\mathbf{b}_i - \mathbf{p}) \times \hat{\mathbf{g}}_i]. \quad (4.8)$$

Replacing expression (4.8) for  $\dot{l}_i$  in (4.7), gives

$$\dot{\mathbf{p}} \cdot \mathbf{s}_i + \omega \cdot [(\mathbf{b}_i - \mathbf{p}) \times \mathbf{s}_i - l_i (\hat{\mathbf{r}}_i \cdot \hat{\mathbf{g}}_i) \hat{\mathbf{h}}_i] = 0, \quad (4.9)$$

where

$$\mathbf{s}_i = \hat{\mathbf{h}}_i \times \hat{\mathbf{r}}_i - [\hat{\mathbf{g}}_i \cdot (\hat{\mathbf{h}}_i \times \hat{\mathbf{r}}_i)] \hat{\mathbf{g}}_i, \quad (4.10)$$

is the component of  $\hat{\mathbf{h}}_i \times \hat{\mathbf{r}}_i$  perpendicular to  $\hat{\mathbf{g}}_i$ .

Finally, rewriting Eqs. (4.8) and (4.9), for  $i = 1, 2, 3$ , in matrix form yields

$$\begin{pmatrix} \mathbf{I}_{3 \times 3} \\ \mathbf{O}_{3 \times 3} \end{pmatrix} \dot{\mathbf{i}} = \begin{pmatrix} \mathbf{G}_{3 \times 3} & \mathbf{K}_{3 \times 3} \\ \mathbf{S}_{3 \times 3} & \mathbf{J}_{3 \times 3} \end{pmatrix} \begin{pmatrix} \dot{\mathbf{p}} \\ \boldsymbol{\omega} \end{pmatrix}, \quad (4.11)$$

where  $\mathbf{I}_{3 \times 3}$  and  $\mathbf{O}_{3 \times 3}$  are the  $3 \times 3$  identity and zero matrix, respectively,  $\dot{\mathbf{i}} = (\dot{l}_1, \dot{l}_2, \dot{l}_3)$  is the vector collecting the joint rates of the actuated joints, and

$$\mathbf{K}^T[i, :] = (\mathbf{b}_i - \mathbf{p}) \times \hat{\mathbf{g}}_i, \quad (4.12)$$

$$\mathbf{G}^T[i, :] = \hat{\mathbf{g}}_i, \quad (4.13)$$

$$\mathbf{J}^T[i, :] = (\mathbf{b}_i - \mathbf{p}) \times \mathbf{s}_i - l_i(\hat{\mathbf{r}}_i \cdot \hat{\mathbf{g}}_i)\hat{\mathbf{h}}_i, \quad (4.14)$$

$$\mathbf{S}^T[i, :] = \mathbf{s}_i, \quad (4.15)$$

with the notation  $\mathbf{A}[i, :]$  to mean the  $i$ th row of matrix  $\mathbf{A}_{3 \times 3}$ .

The matrix relationship (4.11) represents the sought-after input-output kinematics of a 3S<sub>n</sub>PU robot.

### 4.3 Statics Analysis

The input-output static relationship can be immediately deduced from (4.11) through the principle of virtual work. Nevertheless, in order to find out how the loads act upon the legs and are transmitted through the joints, the complete static analysis of the 3S<sub>n</sub>PU will be next carried out.

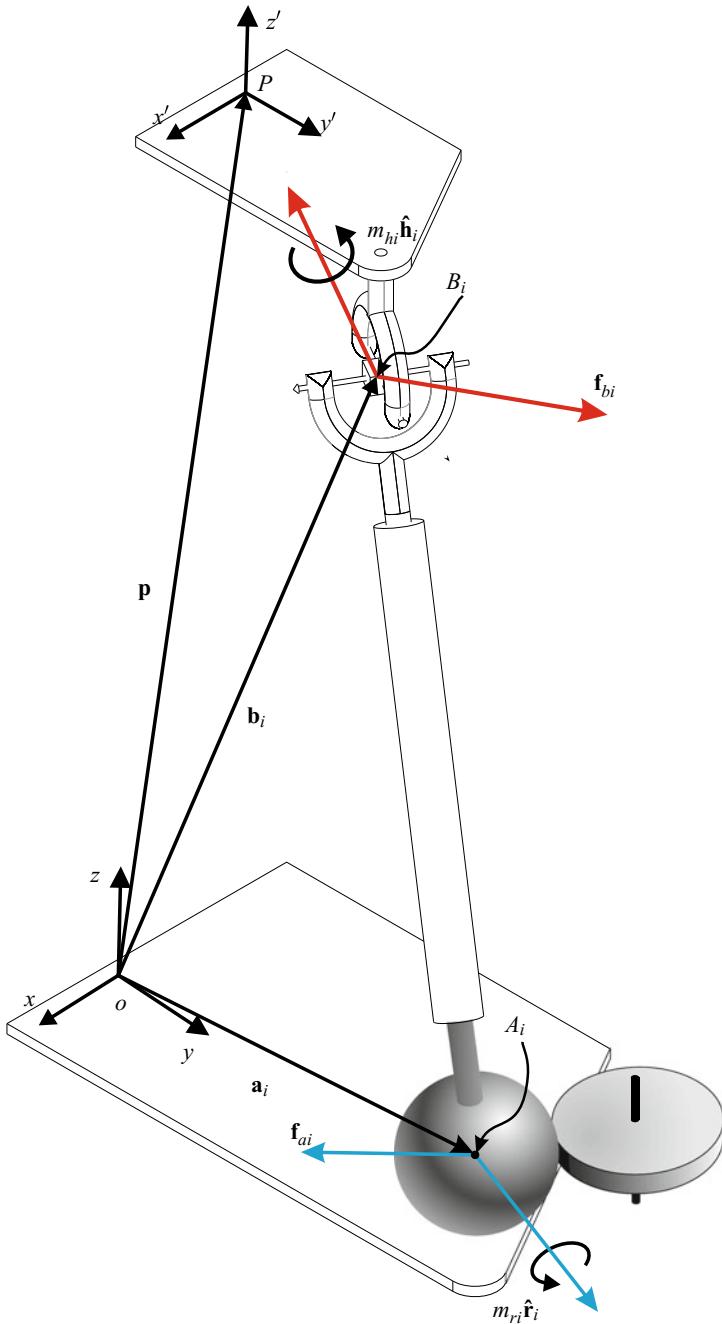
Figure 4.3 shows the free-body diagram of the  $i$ th leg. With reference to Fig. 4.3, the force  $\mathbf{f}_{bi}$  ( $\mathbf{f}_{ai}$ ), applied on  $B_i$  ( $A_i$ ), together with the torque  $m_{hi}\hat{\mathbf{h}}_i$  ( $m_{ri}\hat{\mathbf{r}}_i$ ) are the resultants of the constraint forces exerted by the moving platform (fixed base) on the  $i$ th leg through the U joint (the S<sub>n</sub> joint). Moreover, the force  $\mathbf{f}_{ext}$ , applied on point  $P$  of the moving platform, together with the torque  $\mathbf{m}_{ext}$  will denote the resultants of the interaction forces exerted on the moving platform. The force  $\tau_i\hat{\mathbf{g}}_i$  will denote the axial force exerted on the upper part of the  $i$ th leg by the actuator in the prismatic joint. It is worth noting that the force equilibrium, along the leg axis, of the the upper part of the  $i$ th leg yields the following relationship  $\tau_i = \mathbf{f}_{bi} \cdot \hat{\mathbf{g}}_i$ .

With the introduced notation, the equilibrium of the forces applied on the  $i$ th leg yields  $\mathbf{f}_{bi} + \mathbf{f}_{ai} = 0$ ; whereas, taking  $A_i$  as reference point, the equilibrium of the moments applied on the same leg is:

$$m_{hi}\hat{\mathbf{h}}_i + m_{ri}\hat{\mathbf{r}}_i + l_i\hat{\mathbf{g}}_i \times \mathbf{f}_{bi} = 0. \quad (4.16)$$

The dot product of Eq. (4.16) by  $\hat{\mathbf{g}}_i$ , yields the relationship

$$m_{ri} = -m_{hi} \frac{\hat{\mathbf{h}}_i \cdot \hat{\mathbf{g}}_i}{\hat{\mathbf{r}}_i \cdot \hat{\mathbf{g}}_i}. \quad (4.17)$$



**Fig. 4.3** Free-body diagram associated with the  $i$ -th leg of the studied 3S<sub>n</sub>PU robot

The substitution of the above expression for  $m_{ri}$  in (4.16), leads to

$$\frac{m_{hi}}{\hat{\mathbf{r}}_i \cdot \hat{\mathbf{g}}_i} \left( (\hat{\mathbf{r}}_i \cdot \hat{\mathbf{g}}_i) \hat{\mathbf{h}}_i - (\hat{\mathbf{h}}_i \cdot \hat{\mathbf{g}}_i) \hat{\mathbf{r}}_i \right) + l_i \hat{\mathbf{g}}_i \times \mathbf{f}_{bi} = \frac{m_{hi}}{\hat{\mathbf{r}}_i \cdot \hat{\mathbf{g}}_i} \left( \hat{\mathbf{g}}_i \times (\hat{\mathbf{h}}_i \times \hat{\mathbf{r}}_i) \right) + l_i \hat{\mathbf{g}}_i \times \mathbf{f}_{bi} = \mathbf{0}, \quad (4.18)$$

where the vector identity  $\hat{\mathbf{g}}_i \times (\hat{\mathbf{h}}_i \times \hat{\mathbf{r}}_i) = (\hat{\mathbf{r}}_i \cdot \hat{\mathbf{g}}_i) \hat{\mathbf{h}}_i - (\hat{\mathbf{h}}_i \cdot \hat{\mathbf{g}}_i) \hat{\mathbf{r}}_i$  has been used.

The dot product of Eq. (4.18) by  $\hat{\mathbf{h}}_i \times \hat{\mathbf{r}}_i$  yields the relationship:

$$(\hat{\mathbf{g}}_i \times \mathbf{f}_{bi}) \cdot (\hat{\mathbf{h}}_i \times \hat{\mathbf{r}}_i) = 0, \quad (4.19)$$

which is satisfied if, and only if,  $\mathbf{f}_{bi}$  is a linear combination of  $\hat{\mathbf{g}}_i$  and  $\hat{\mathbf{h}}_i \times \hat{\mathbf{r}}_i$ . Subtracting from  $\hat{\mathbf{h}}_i \times \hat{\mathbf{r}}_i$  its component along  $\hat{\mathbf{g}}_i$ ,  $\mathbf{s}_i$  is obtained. Since  $\hat{\mathbf{g}}_i$  and  $\mathbf{s}_i$  are two orthogonal vectors that span the same subspace as  $\hat{\mathbf{g}}_i$  and  $\hat{\mathbf{h}}_i \times \hat{\mathbf{r}}_i$ ,  $\mathbf{f}_{bi}$  can be expressed as follows:

$$\mathbf{f}_{bi} = \tau_i \hat{\mathbf{g}}_i + \tau_i^\perp \mathbf{s}_i. \quad (4.20)$$

Equation (4.20) can be interpreted as the equilibrium of the forces applied on the upper part of the  $i$ th leg. In fact, the two forces  $\tau_i \hat{\mathbf{g}}_i$  and  $\tau_i^\perp \mathbf{s}_i$  are the active axial and the passive shear forces applied through the actuated prismatic joint, respectively.

Replacing expression (4.20) for  $\mathbf{f}_{bi}$  in (4.18), and realizing that  $\hat{\mathbf{g}}_i \times (\hat{\mathbf{h}}_i \times \hat{\mathbf{r}}_i) = \hat{\mathbf{g}}_i \times \mathbf{s}_i$  yields

$$\left( \frac{m_{hi}}{\hat{\mathbf{r}}_i \cdot \hat{\mathbf{g}}_i} + l_i \tau_i^\perp \right) (\hat{\mathbf{g}}_i \times \mathbf{s}_i) = \mathbf{0},$$

which is satisfied if

$$m_{hi} = -l_i \tau_i^\perp (\hat{\mathbf{r}}_i \cdot \hat{\mathbf{g}}_i). \quad (4.21)$$

Using Eq. (4.17), Eq. (4.21) can be rewritten as

$$m_{ri} = l_i \tau_i^\perp (\hat{\mathbf{h}}_i \cdot \hat{\mathbf{g}}_i). \quad (4.22)$$

Regarding the equilibrium of the forces at the moving platform, we have that:

$$\mathbf{f}_{ext} = - \sum_{i=1}^3 \mathbf{f}_{bi} = - \sum_{i=1}^3 \tau_i \hat{\mathbf{g}}_i - \sum_{i=1}^3 \tau_i^\perp \mathbf{s}_i, \quad (4.23)$$

and, taking point  $P$  in the moving platform as reference point, the equilibrium of moments is:

$$\mathbf{m}_{ext} = - \sum_{i=1}^3 m_{hi} \hat{\mathbf{h}}_i - \sum_{i=1}^3 (\mathbf{b}_i - \mathbf{p}) \times \mathbf{f}_{bi}. \quad (4.24)$$

Substituting  $\mathbf{f}_{bi}$ , according to (4.20), and  $m_{hi}$ , according to (4.21), yields

$$\mathbf{m}_{ext} = - \sum_{i=1}^3 \tau_i (\mathbf{b}_i - \mathbf{p}) \times \hat{\mathbf{g}}_i - \sum_{i=1}^3 \tau_i^\perp \left( (\mathbf{b}_i - \mathbf{p}) \times \mathbf{s}_i - l_i (\hat{\mathbf{r}}_i \cdot \hat{\mathbf{g}}_i) \hat{\mathbf{h}}_i \right). \quad (4.25)$$

Finally, Eqs.(4.23) and (4.25),  $i = 1, 2, 3$ , can be rewritten in matrix form as follows:

$$\begin{pmatrix} \mathbf{f}_{ext} \\ \mathbf{m}_{ext} \end{pmatrix} = - \begin{pmatrix} \mathbf{G}_{3 \times 3} & \mathbf{K}_{3 \times 3} \\ \mathbf{S}_{3 \times 3} & \mathbf{J}_{3 \times 3} \end{pmatrix}^T \begin{pmatrix} \boldsymbol{\tau} \\ \boldsymbol{\tau}^\perp \end{pmatrix}, \quad (4.26)$$

where  $\boldsymbol{\tau} = (\tau_1, \tau_2, \tau_3)$  is a vector collecting the signed magnitudes of forces applied by the actuators in the prismatic joints, whereas  $\boldsymbol{\tau}^\perp = (\tau_1^\perp, \tau_2^\perp, \tau_3^\perp)$  collects the passive shear forces.

Finally, matrix relationship (4.26) is the input-output static relationship of the 3S<sub>n</sub>PU robot. It is worth noting that (4.11) and (4.26) satisfy the instantaneous power balance:  $\mathbf{f}_{ext} \cdot \dot{\mathbf{p}} + \mathbf{m}_{ext} \cdot \boldsymbol{\omega} = -\boldsymbol{\tau} \cdot \dot{\mathbf{i}}$ .

## 4.4 Singularities

Singularities are robot configurations where the relationship between the rates of the actuated joints variables and the moving platform twist,  $(\dot{\mathbf{p}}, \boldsymbol{\omega})$ , is not one-to-one [4, 14]. In any parallel robot, there are two basic kinds of singularities [4]: singularities of the inverse kinematic problem and singularities of the direct kinematic problem. While the former occur when the actuated joint rates cannot be uniquely computed for an assigned moving platform twist, the latter occur when the moving platform twist cannot be uniquely determined for assigned actuated joint rates.

Regarding the singularities of the inverse kinematic problem, we can always obtain valid values for  $\dot{\mathbf{i}}$  for an assigned twist for the moving platform,  $(\dot{\mathbf{p}}, \boldsymbol{\omega})$ , provided that the last three equations of system (4.11) are satisfied. Nevertheless, note that system (4.11) does not model the mobility limitations of the real joints. Such limitations bound the workspace and, when correctly modeled, also yield to singularities of the inverse kinematic problem.

Concerning the singularities of the direct kinematics, observe that Eq. (4.11) can be rewritten as follows:

$$\dot{\mathbf{i}} = \underbrace{(\mathbf{G}_{3 \times 3} \ \mathbf{K}_{3 \times 3})}_{\mathbf{J}_x} \begin{pmatrix} \dot{\mathbf{p}} \\ \boldsymbol{\omega} \end{pmatrix}, \quad (4.27)$$

$$\mathbf{0}_{3 \times 1} = \underbrace{(\mathbf{S}_{3 \times 3} \ \mathbf{J}_{3 \times 3})}_{\mathbf{J}_c} \begin{pmatrix} \dot{\mathbf{p}} \\ \boldsymbol{\omega} \end{pmatrix}. \quad (4.28)$$

Now, Eq. (4.27) relates the twist of the moving platform with the linear velocities of the prismatic actuators. If  $\mathbf{J}_x$ —sometimes called *Jacobian of actuations*—is rank-

deficient for all possible locations of the moving platform, the robot is said to be *architecturally singular*. Observe how  $\mathbf{J}_x$  is independent on how the U joints are arranged; it only depends on the three leg lines.

Equation (4.28) models the internal constraints. If  $\mathbf{J}_c$ —sometimes called *Jacobian of constraints*—is rank-deficient, there are non-null twists for the moving platform that satisfy (4.28). When this happens, the robot is said to be in a *constraint singularity*.

In general, the singularities of the direct kinematics correspond to those configurations in which:

$$\det \begin{pmatrix} \mathbf{G}_{3 \times 3} & \mathbf{K}_{3 \times 3} \\ \mathbf{S}_{3 \times 3} & \mathbf{J}_{3 \times 3} \end{pmatrix} = 0, \quad (4.29)$$

which is known as the singularity locus. Unfortunately, the geometric interpretation of the above algebraic condition is not straightforward. Nevertheless, (4.28) allows the elimination of  $\dot{\mathbf{p}}$  provided that  $\det(\mathbf{S}_{3 \times 3}) = \mathbf{s}_1 \cdot (\mathbf{s}_2 \times \mathbf{s}_3)$  is different from zero. In this case, system (4.11) becomes

$$\dot{\mathbf{i}} = \mathbf{Q}\omega, \quad (4.30)$$

where  $\mathbf{Q}$  is the  $3 \times 3$  matrix  $(\mathbf{K}_{3 \times 3} - \mathbf{G}_{3 \times 3}\mathbf{S}_{3 \times 3}^{-1}\mathbf{J}_{3 \times 3})$ . Thus, the analytic expression of the singularity locus becomes

$$\det(\mathbf{Q}_{3 \times 3}) = \mathbf{q}_1 \cdot (\mathbf{q}_2 \times \mathbf{q}_3) = 0, \quad (4.31)$$

where  $\mathbf{q}_i$ ,  $i = 1, 2, 3$ , are the column vectors of  $\mathbf{Q}$ . In conclusion, if the triple product  $\mathbf{s}_1 \cdot (\mathbf{s}_2 \times \mathbf{s}_3)$  is different from zero (i.e., the three vectors  $\mathbf{s}_i$ ,  $i = 1, 2, 3$ , are neither coplanar nor none of them is a null vector), the singularities of the forward kinematics are geometrically identified by either the coplanarity of the three vectors  $\mathbf{q}_i$ ,  $i = 1, 2, 3$ , or by the fact that at least one of these vectors is a null vector.

If the triple product  $\mathbf{s}_1 \cdot (\mathbf{s}_2 \times \mathbf{s}_3)$  is zero, the determinant of the whole  $6 \times 6$  matrix appearing in (4.29) must be considered, and geometric interpretations of (4.29) are much more difficult to provide. The triple product  $\mathbf{s}_1 \cdot (\mathbf{s}_2 \times \mathbf{s}_3)$  vanishes when either: (a) at least one of the  $\mathbf{s}_i$  vectors is a null vector, or (b) the three  $\mathbf{s}_i$  vectors are coplanar. Since  $\mathbf{s}_i$  is the component of  $\hat{\mathbf{h}}_i \times \hat{\mathbf{r}}_i$  perpendicular to  $\hat{\mathbf{g}}_i$  (i.e., to the leg axis), (a) occurs, in at least one leg, when either:

- (a.1) the two unit vectors  $\hat{\mathbf{h}}_i$  and  $\hat{\mathbf{r}}_i$  are parallel (i.e., when, in a leg, the revolute joint axes in the U joint are both parallel to the plane defined by the roller axis and the sphere center in the S<sub>n</sub> joint), or
- (a.2) the leg axis is the intersection line between the plane, defined by the roller axis and the sphere center in the S<sub>n</sub> joint, and the plane, defined by the revolute pair axes of the U joint.

Condition (a.2) is forbidden in practice by the actual sizes of joints and links. Regarding condition (a.1), a very special case occurs when  $\hat{\mathbf{h}}_i$  and  $\hat{\mathbf{r}}_i$  are parallel in the three legs. This makes  $\mathbf{S}_{3 \times 3}$  to become a null matrix and allows the determinant

at the left-hand side of (4.29) to be factorized as  $\det(\mathbf{G}_{3 \times 3})\det(\mathbf{J}_{3 \times 3})$  where

$$\det(\mathbf{G}_{3 \times 3}) = \hat{\mathbf{g}}_1 \cdot (\hat{\mathbf{g}}_2 \times \hat{\mathbf{g}}_3), \quad (4.32)$$

and

$$\det(\mathbf{J}_{3 \times 3}) = -l_1 l_2 l_3 (\hat{\mathbf{r}}_1 \cdot \hat{\mathbf{g}}_1)(\hat{\mathbf{r}}_2 \cdot \hat{\mathbf{g}}_2)(\hat{\mathbf{r}}_3 \cdot \hat{\mathbf{g}}_3) \hat{\mathbf{h}}_1 \cdot (\hat{\mathbf{h}}_2 \times \hat{\mathbf{h}}_3). \quad (4.33)$$

Thus, in this case, a singularity occurs when either the leg axes are all parallel to a unique plane, or the  $\hat{\mathbf{h}}_i$  vectors are coplanar, or, finally, in at least one leg, the leg axis lies on the plane defined by the roller axis and the sphere center of the  $S_n$  joint. Moreover, it is worth noting that, in this case, the moving platform performs an instantaneous translation, if neither  $\det(\mathbf{G}_{3 \times 3})$  nor  $\det(\mathbf{J}_{3 \times 3})$  are equal to zero.

Regarding condition (b) (i.e., the coplanarity of the  $\mathbf{s}_i$  vectors), it occurs when the leg axes are all parallel, and in other configurations more difficult to visualize.

## 4.5 Controllability

The pose of the moving platform can be modeled as a point in  $R^3 \times SO(3)$  which is locally diffeomorphic to  $R^6$  equipped with a proper set of local coordinates:  $\mathbf{x} = (\mathbf{p}^T, \eta^T)^T$  where  $\eta$  is a three-dimensional vector collecting the values of the three orientation parameters chosen to define the moving platform orientation.

By using the orientation parameters' rates,  $\dot{\eta}$ , the moving platform angular velocity,  $\omega$ , can be expressed as:

$$\omega = \mathbf{H}_{3 \times 3} \dot{\eta}. \quad (4.34)$$

This relationship allows system (4.11) to be rewritten as

$$\dot{\mathbf{x}} = \mathbf{V}_{6 \times 6} \begin{pmatrix} \mathbf{i} \\ \mathbf{O}_{3 \times 1} \end{pmatrix}, \quad (4.35)$$

with

$$\mathbf{V}_{6 \times 6} \triangleq (\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4, \mathbf{v}_5, \mathbf{v}_6) = \begin{pmatrix} \mathbf{I}_{3 \times 3} & \mathbf{O}_{3 \times 3} \\ \mathbf{O}_{3 \times 3} & \mathbf{H}_{3 \times 3}^{-1} \end{pmatrix} \begin{pmatrix} \mathbf{G}_{3 \times 3} & \mathbf{K}_{3 \times 3} \\ \mathbf{S}_{3 \times 3} & \mathbf{J}_{3 \times 3} \end{pmatrix}^{-1}, \quad (4.36)$$

where  $\mathbf{v}_i$  is the  $i$ th 6-dimensional column vector of matrix  $\mathbf{V}_{6 \times 6}$ . The vectors  $\mathbf{v}_i$  depend only on  $\mathbf{x}$ ; so, they are vector fields defined on moving platform configuration space.

Definition (4.36) allows (4.35) to be further simplified as follows:

$$\dot{\mathbf{x}} = \mathbf{v}_1 \dot{l}_1 + \mathbf{v}_2 \dot{l}_2 + \mathbf{v}_3 \dot{l}_3. \quad (4.37)$$

This relationship states that, in the neighborhood of a generic configuration, say  $\mathbf{x}_0$ , all the configurations,  $\mathbf{x}$ , reachable without maneuvering (i.e., without a sequence of coordinated actions of the actuators) are so located that  $(\mathbf{x} - \mathbf{x}_0) \in \text{Span}(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3)$ .

The key relationship to determine the ability of the moving platform to reach any configuration in the neighborhood of a generic configuration,  $\mathbf{x}_0$ , is (4.37) [2, 9]. The presence of non-holonomic constraints in the 3S<sub>n</sub>PU robot might allow all neighboring configurations be reachable possibly by maneuvering. If this happens, the system would be *locally controllable* [2] at  $\mathbf{x}_0$ .

It can be shown (see [9, pp. 323–324]) that, if a system, satisfying (4.37) at the configuration  $\mathbf{x}_0$ , first follows  $\mathbf{v}_i$ ,  $i \in \{1, 2, 3\}$ , for a small time  $\varepsilon$ , then follows  $\mathbf{v}_j$ ,  $j \in \{1, 2, 3 : j \neq i\}$ , for the same time  $\varepsilon$ , then  $-\mathbf{v}_i$  for  $\varepsilon$ , and finally  $-\mathbf{v}_j$  for  $\varepsilon$ , it will reach the following configuration:

$$\lim_{\varepsilon \rightarrow 0} \mathbf{x}(4\varepsilon) = \mathbf{x}_0 + \varepsilon^2 [\mathbf{v}_i, \mathbf{v}_j], \quad (4.38)$$

where  $[\mathbf{v}_i, \mathbf{v}_j]$  is the 6-dimensional vector field named Lie product of  $\mathbf{v}_i$  and  $\mathbf{v}_j$ , defined as follows

$$[\mathbf{v}_i, \mathbf{v}_j] = \frac{\partial \mathbf{v}_j}{\partial \mathbf{x}} \mathbf{v}_i - \frac{\partial \mathbf{v}_i}{\partial \mathbf{x}} \mathbf{v}_j, \quad (4.39)$$

and the trailing subscript,  $\mathbf{x} = \mathbf{x}_0$ , indicates the point the two vector fields,  $\mathbf{v}_i$  and  $\mathbf{v}_j$ , are evaluated at.

By reiterating the same reasoning (first, on pairs of vector fields of type  $\mathbf{v}_i$  and  $[\mathbf{v}_j, \mathbf{v}_k]$ ,  $i, j, k \in \{1, 2, 3 : i \neq j \neq k\}$ , and, successively, on pairs of vector fields belonging to the set which collects all the vector fields that, in the previous iterations, were demonstrated to point from  $\mathbf{x}_0$  toward reachable configurations), it can be demonstrated that all the vector fields obtained through Lie products of any degree of elements of the set  $\{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$  point toward configurations that are reachable by maneuvering from  $\mathbf{x}_0$  [2]. In other words, for any reachable configuration, say  $\mathbf{x}$ , the vector ( $\mathbf{x} = \mathbf{x}_0$ ) belongs to the Lie algebra of  $\{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$  (The *Lie algebra* of a set of vector fields is the linear span of all Lie products, of all degrees, of vector fields belonging to that set [2]).

In our case, demonstrating that the dimension of the linear space

$$\text{span}(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, [\mathbf{v}_1, \mathbf{v}_2], [\mathbf{v}_2, \mathbf{v}_3], [\mathbf{v}_3, \mathbf{v}_1]),$$

is six is sufficient for concluding that the robot is locally controllable at a given configuration since configuration space of the moving platform is 6-dimensional (Chow's theorem [9]). It is worth noting that, if the dimension of

$$\text{span}(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, [\mathbf{v}_1, \mathbf{v}_2], [\mathbf{v}_2, \mathbf{v}_3], [\mathbf{v}_3, \mathbf{v}_1]), \quad (4.40)$$

equals six, all the Lie products of any degree in  $\{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$  must belong to this space. In this case, all the reachable configurations,  $\mathbf{x}$ , satisfy the condition

$$(\mathbf{x} - \mathbf{x}_0) \in \text{span}(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, [\mathbf{v}_1, \mathbf{v}_2], [\mathbf{v}_2, \mathbf{v}_3], [\mathbf{v}_3, \mathbf{v}_1]). \quad (4.41)$$

Moreover, showing that the set of configurations where the system is locally controllable is a simply connected region is sufficient to demonstrate the existence of finite regions of the moving platform configuration space where, for any two configurations belonging to that region, at least one path exists, which the system can follow, for moving from one configuration to the other (i.e., the system is *globally controllable* in that region).

According to the above discussion, the configurations where the local controllability of our robot is not guaranteed are the geometric locus of the roots of the following equation

$$\det[\mathbf{L}_{6 \times 6}] = 0, \quad (4.42)$$

where

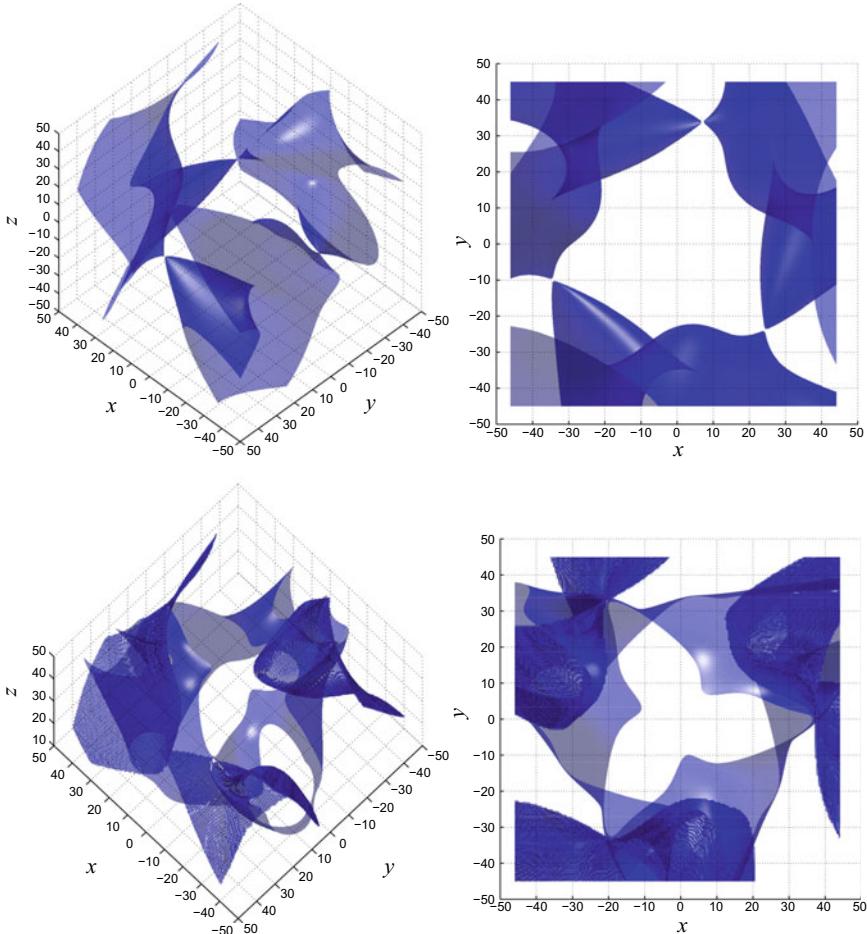
$$\mathbf{L}_{6 \times 6} \triangleq [\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, [\mathbf{v}_1, \mathbf{v}_2], [\mathbf{v}_2, \mathbf{v}_3], [\mathbf{v}_3, \mathbf{v}_1]]. \quad (4.43)$$

The locus of the roots of (4.42) is in general a 5-dimensional variety; thus, a finite region where our robot is globally controllable in general exists. This statement will be verified through the following numerical example.

## 4.6 Example

According to the notation used in the Figs. 4.2 and 4.3, let us consider a 3S<sub>n</sub>PU robot where points  $A_i$  ( $B_i$ ),  $i = 1, 2, 3$ , are at the vertices of an equilateral triangle fixed to the base (to the moving platform). The Cartesian reference system fixed to the base (to the moving platform) has the origin  $O$  ( $P$ ) at the centroid of the equilateral triangle, the  $z$ -axis is perpendicular to the plane of the triangle and, the  $x$ -axis pass through  $A_3$  ( $B_3$ ) with positive direction from  $A_3$  ( $B_3$ ) toward  $O$  ( $P$ ). The distance of the triangle vertices  $A_i$  ( $B_i$ ) from its centroid is set to 39.7 cm (11.76 cm). The geometry of the base and the moving platform are defined by the following data (the vectors without superscript are measured in the base reference frame, whereas those with the superscript  $(\cdot)^e$  are measured in the reference frame of the moving platform):

	Leg 1	Leg 2	Leg 3
$\mathbf{a}_i$	$(19.84, -34.38, 0.0)^T$	$(19.85, 34.38, 0.0)^T$	$(-39.70, 0.0, 0.0)^T$
$\hat{\mathbf{r}}_i$	$(0.353, -0.612, 0.707)^T$	$(0.353, 0.612, 0.707)^T$	$(-0.707, 0.0, 0.707)^T$
$\mathbf{b}_i^e$	$(5.883, -10.19, 0.0)^T$	$(5.884, 10.19, 0.0)^T$	$(-11.76, 0.0, 0.0)^T$
$\mathbf{w}_{4i}^e$	$(0.353, -0.612, 0.707)^T$	$(0.353, 0.612, 0.707)^T$	$(-0.707, 0.0, 0.707)^T$



**Fig. 4.4** Top: Axonometric and zenithal views of the singularity locus, defined by Eq. (4.29). Bottom: Axonometric and zenithal views of the region, defined by Eq. (4.42), where the robot is not controllable. In both cases the moving platform is translated in the xyz-space from  $-50$  to  $50$  cm, while keeping its orientation fixed to  $(0, 1, 0)$  radians, using XZX Euler angles

With this robot geometry, the singularity locus defined by Eq. (4.29) and the root locus of Eq. (4.42) have been computed, for a fixed orientation of the moving platform, with respect to the base reference frame. The results of these two computations are shown in the Fig. 4.4 (top) and 4.4 (bottom), respectively. By integrating the singularity locus and the root locus of Eq. (4.42), an ample region free from singularities where the robot is also globally controllable is identified. This information is needed both in the design of this robot and in its control. It will be the base for the motion planning algorithm presented in the next chapter.

## References

1. P. Ben-Horin, F. Thomas, A nonholonomic 3-DoF parallel robot, *Advances in Robot Kinematics: Analysis and Design* (Springer, Berlin, 2008), pp. 111–118
2. H. Choset, K. M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. E. Kavraki, S. Thrun, *Principles of Robot Motion: Theory, Algorithms, and Implementations*, Intelligent Robotics and Autonomous Agents Series (The MIT Press, Cambridge, 2005)
3. N. Farhat, V. Mata, A. Page, F. Valero, Identification of dynamic parameters of a 3-DoF RPS parallel manipulator. *Mech. Mach. Theory* **43**(1), 1–17 (2008)
4. C. Gosselin, J. Angeles, Singularity analysis of closed-loop kinematic chains. *IEEE Trans. Robot. Autom.* **6**(3), 281–290 (1990)
5. P. Grosch, R. Di Gregorio, F. Thomas, generation of under-actuated parallel robots with non-holonomic articulations and kinetostatic analysis of a case-study, in *Proceedings of the ASME International Design Engineering Technical Conference and Computers and Information in Engineering Conference*, San Diego, USA (2009), pp. 979–986
6. P. Grosch, R. Di Gregorio, F. Thomas, Generation of under-actuated manipulators with non-holonomic articulations from ordinary manipulators. *J. Mech. Robot.* **2**(1), 11005–11012 (2010)
7. C. Innocenti, V. Parenti-Castelli, Direct position analysis of the Stewart platform mechanism. *Mech. Mach. Theory* **25**(6), 611–621 (1990)
8. C.Y. Ji, T.C. Chen, Y.L. Lee, Investigation of kinematic analysis and applications for a 3-RRPS parallel manipulator. *J. Chin. Soc. Mech. Eng.* **28**(6), 623–632 (2007)
9. R.M. Murray, Z. Li, S.S. Sastry, *A Mathematical Introduction to Robotic Manipulation* (CRC Press, Boca Raton, 1994)
10. V. Parenti-Castelli, R. Di Gregorio, Closed-form solution of the direct kinematics of the 6-3 type Stewart platform using one extra sensor. *Meccanica* **31**(6), 705–711 (1996)
11. N.M. Rao, K.M. Rao, Dimensional synthesis of a spatial 3-RPS parallel manipulator for a prescribed range of motion of spherical joints. *Mech. Mach. Theory* **44**(2), 477–486 (2009)
12. A. Sokolov, P. Xirouchakis, Dynamics analysis of a 3-DOF parallel manipulator with RPS joint structure. *Mech. Mach. Theory* **42**(5), 541–557 (2007)
13. D. Stewart, A platform with six degrees of freedom. *Proc. Inst. Mech. Eng.* **180**(1), 371–378 (1965)
14. D. Zlatanov, R.G. Fenton, B. Benabib, A unifying framework for classification and interpretation of mechanism singularities. *J. Mech. Des.* **117**(4), 566–572 (1995)
15. D. Zlatanov, L.A. Bonev, C. Gosselin, Constraint singularities of parallel mechanisms, in *IEEE International Conference on Robotics and Automation ICRA* (2002), pp. 496–502

# Chapter 5

## Motion Planning for the $3S_nPU$ Robot



The kinetostatics of the  $3S_nPU$  robot allowed us, in the previous chapter, to prove that this robot is able to locally move its moving platform—within some regions—in a six-dimensional configuration space. In this chapter we go a step further by presenting a solution to the motion planning problem which can be adapted to other non-holonomic parallel robots. This chapter presents some results obtained in cooperation with professors Krzysztof Tchoń and Janusz Jakubiak from Wrocław University of Technology, Poland, which already appeared in [3].

The motion planning problem for the  $3S_nPU$  robot is here addressed using the endogenous configuration space approach [2], specified in [1] for a class of mechanical systems that includes this parallel non-holonomic robot. The solution to this motion planning problem is decomposed into two steps: first the control system representing the robot's kinematics is subject to a feedback transformation, and then the end-point map of the obtained system is inverted.

This chapter is simply organized as follows: Sect. 5.1 introduces the motion planning algorithm whose performance is illustrated in Sect. 5.2 by a numeric example.

### 5.1 Motion Planning

In the previous chapter, it is proven that the instantaneous kinematics of the  $3S_nPU$  robot can be expressed in a very compact way as

$$\begin{pmatrix} \mathbf{I}_{3 \times 3} \\ \mathbf{O}_{3 \times 3} \end{pmatrix} \dot{\mathbf{i}} = \begin{pmatrix} \mathbf{G}_{3 \times 3} & \mathbf{K}_{3 \times 3} \\ \mathbf{S}_{3 \times 3} & \mathbf{J}_{3 \times 3} \end{pmatrix} \begin{pmatrix} \dot{\mathbf{p}} \\ \boldsymbol{\omega} \end{pmatrix}, \quad (5.1)$$

where  $\dot{\mathbf{i}} = (\dot{i}_1, \dot{i}_2, \dot{i}_3)^T$  is the vector of velocities in the actuators,  $\begin{pmatrix} \dot{\mathbf{p}} \\ \boldsymbol{\omega} \end{pmatrix}$  is the vector of linear and angular velocities of the moving platform,  $\mathbf{I}_{3 \times 3}$  and  $\mathbf{O}_{3 \times 3}$  are the  $3 \times 3$  identity and the zero matrix, respectively, and  $\mathbf{G} = \mathbf{G}(\mathbf{p}, \mathbf{R})$ ,  $\mathbf{K} = \mathbf{K}(\mathbf{p}, \mathbf{R})$ ,  $\mathbf{S} =$

$\mathbf{S}(\mathbf{p}, \mathbf{R})$ ,  $\mathbf{J} = \mathbf{J}(\mathbf{p}, \mathbf{R})$  are  $3 \times 3$  matrices dependent on the position and orientation (pose) of the moving platform  $(\mathbf{p}, \mathbf{R}) \in \mathbb{R}^3 \times SO(3)$  whose entries are defined as (see Chap. 4 for details)

$$\mathbf{K}^T[i, :] = (\mathbf{b}_i - \mathbf{p}) \times \hat{\mathbf{g}}_i, \quad (5.2)$$

$$\mathbf{G}^T[i, :] = \hat{\mathbf{g}}_i, \quad (5.3)$$

$$\mathbf{J}^T[i, :] = (\mathbf{b}_i - \mathbf{p}) \times \mathbf{s}_i - l_i(\hat{\mathbf{r}}_i \cdot \hat{\mathbf{g}}_i)\hat{\mathbf{h}}_i, \quad (5.4)$$

$$\mathbf{S}^T[i, :] = \mathbf{s}_i, \quad (5.5)$$

where  $\mathbf{A}[i, :]$  denotes the  $i$ th row of a matrix  $\mathbf{A}$ .

If the robot is in a singularity-free region, the  $6 \times 6$  matrix in (5.1) can be inverted. Then, we can define

$$\begin{pmatrix} \mathbf{E}(\mathbf{p}, \mathbf{R}) \\ \mathbf{F}(\mathbf{p}, \mathbf{R}) \end{pmatrix} = \begin{pmatrix} \mathbf{G} & \mathbf{K} \\ \mathbf{S} & \mathbf{J} \end{pmatrix}^{-1} \Big|_{\text{3 first columns}}. \quad (5.6)$$

Now, taking  $\mathbf{u} = \dot{\mathbf{i}}$  as a control variable, the kinematics model can be converted into the driftless system (no motion takes place for null inputs)

$$\begin{cases} \dot{\mathbf{p}} = \mathbf{E}(\mathbf{p}, \mathbf{R})\mathbf{u} \\ \dot{\mathbf{R}} = [\mathbf{F}(\mathbf{p}, \mathbf{R})\mathbf{u}]_\times \mathbf{R}, \end{cases} \quad (5.7)$$

where  $[\cdot]_\times : \mathbb{R}^3 \rightarrow so(3)$  denotes the standard Lie algebra isomorphism of  $\mathbb{R}^3$  with the cross product and the space of skew symmetric  $3 \times 3$  matrices with the matrix commutator, so that  $[\mathbf{v} \times \mathbf{w}]_\times = [\mathbf{v}]_\times[\mathbf{w}]_\times - [\mathbf{w}]_\times[\mathbf{v}]_\times = [[\mathbf{v}]_\times, [\mathbf{w}]_\times]$ .

Given the control system represented by (5.7), the motion planning problem for the 3S<sub>n</sub>PU parallel non-holonomic robot can be stated in the following way: compute a control function  $\mathbf{u}(t)$  steering the system from an initial moving platform pose, say  $(\mathbf{p}_0, \mathbf{R}_0)$ , to the desired one, say  $(\mathbf{p}_d, \mathbf{R}_d)$ , within a prescribed time  $T$ . More formally, setting  $\mathbf{p}(t) = \mathbf{p}_{\mathbf{p}_0, \mathbf{R}_0, t}(\mathbf{u}(\cdot))$ ,  $\mathbf{R}(t) = \mathbf{R}_{\mathbf{p}_0, \mathbf{R}_0, t}(\mathbf{u}(\cdot))$  to be the trajectory of (5.7) starting at  $(\mathbf{p}_0, \mathbf{R}_0)$  and driven by the control  $\mathbf{u}(t)$ , this means that at time  $T$  the system reaches the prescribed values  $\mathbf{p}(T) = \mathbf{p}_d$  and  $\mathbf{R}(T) = \mathbf{R}_d$ .

Due to the complexity of the entries of the  $6 \times 6$  matrix in (5.1), the analytic forms of  $\mathbf{E}$  and  $\mathbf{F}$  are not very enlightening. Nevertheless, if we assume that  $\mathbf{G}$  is invertible, (5.6) can be rewritten as:

$$\begin{pmatrix} \mathbf{E}(\mathbf{p}, \mathbf{R}) \\ \mathbf{F}(\mathbf{p}, \mathbf{R}) \end{pmatrix} = \begin{pmatrix} \mathbf{G}^{-1} + \mathbf{G}^{-1}\mathbf{K}(\mathbf{J} - \mathbf{S}\mathbf{G}^{-1}\mathbf{K})^{-1}\mathbf{S}\mathbf{G}^{-1} \\ -(\mathbf{J} - \mathbf{S}\mathbf{G}^{-1}\mathbf{K})^{-1}\mathbf{S}\mathbf{G}^{-1} \end{pmatrix}. \quad (5.8)$$

If we also assume that  $\mathbf{S}$  is invertible, the following feedback

$$\mathbf{u} = \mathbf{G}\mathbf{S}^{-1}(\mathbf{J} - \mathbf{S}\mathbf{G}^{-1}\mathbf{K})\mathbf{v}, \quad (5.9)$$

where  $\mathbf{v} \in \mathbb{R}^3$  is a new control, makes the control system given by (5.7) equivalent to

$$\begin{cases} \dot{\mathbf{p}} = \mathbf{S}^{-1} \mathbf{J} \mathbf{v} \\ \dot{\mathbf{R}} = -[\mathbf{v}]_{\times} \mathbf{R}. \end{cases} \quad (5.10)$$

Therefore, if both  $\mathbf{G}$  and  $\mathbf{S}$  are invertible, the solution of the motion planning problem may be obtained in two steps: (1) a control  $\mathbf{v}(t)$  solving the motion planning problem for the system (5.10) is found, and (2) the original control  $\mathbf{u}(t)$  is computed using (5.9).

The first step can be accomplished in accordance with the guidelines presented in [1], that is concisely summarized below.

Let  $\mathbf{v}_\theta(t)$  be a family of control functions smoothly dependent on a parameter  $\theta \in \mathbb{R}$ , and let

$$\begin{cases} \mathbf{p}_t(\theta) = \mathbf{p}_{\mathbf{p}_0, \mathbf{R}_0, t}(\mathbf{v}_\theta(\cdot)) \\ \mathbf{R}_t(\theta) = \mathbf{R}_{\mathbf{p}_0, \mathbf{R}_0, t}(\mathbf{v}_\theta(\cdot)), \end{cases} \quad (5.11)$$

denote the trajectory of the system (5.10) initialized at  $(\mathbf{p}_0, \mathbf{R}_0)$  and subject to the control  $\mathbf{v}_\theta(t)$ .

The derivation of the motion planning algorithm for the system (5.10) relies on an assumption that there exists a control family  $\mathbf{v}_\theta(t)$ , such that the error

$$\mathbf{e}(\theta) = (\mathbf{p}_T(\theta) - \mathbf{p}_d, \log(\mathbf{R}_T(\theta)) \mathbf{R}_d^T), \quad (5.12)$$

decreases to zero exponentially as a function of  $\theta$  with a prescribed decay rate  $\gamma > 0$ ,

$$\frac{d\mathbf{e}(\theta)}{d\theta} = -\gamma \mathbf{e}(\theta). \quad (5.13)$$

The logarithm of the rotation matrix in (5.12) is defined as

$$\log \mathbf{R} = \frac{\alpha}{2 \sin \alpha} (\mathbf{R} - \mathbf{R}^T),$$

where

$$\cos \alpha = \frac{1}{2} (\text{Tr } \mathbf{R} - 1),$$

and the angle of rotation  $0 \leq \alpha < \pi$ .

To proceed, we need to introduce the following auxiliary variables:

$$\mathbf{w}_t(\theta) = \frac{\partial \mathbf{p}_t(\theta)}{\partial \theta}, \quad (5.14)$$

$$[\mathbf{s}_t(\theta)]_{\times} = \frac{\partial \mathbf{R}_t(\theta)}{\partial \theta} \mathbf{R}_t^T(\theta). \quad (5.15)$$

The differentiation with respect to  $\theta$  of the matrices on the right hand side of the system (5.10) results in a collection of differential equations (for details see [1], proof of Theorem 2.1) which can be expressed in matrix form as follows:

$$\begin{pmatrix} \dot{\mathbf{w}}_t(\theta) \\ \dot{\mathbf{s}}_t(\theta) \end{pmatrix} = \begin{pmatrix} \mathbf{A}_{11\theta}(t) & \mathbf{A}_{12\theta}(t) \\ \mathbf{0} & -[\mathbf{v}_\theta(t)]_\times \end{pmatrix} \begin{pmatrix} \mathbf{w}_t(\theta) \\ \mathbf{s}_t(\theta) \end{pmatrix} + \begin{pmatrix} \mathbf{B}_{1\theta}(t) \\ -\mathbf{I} \end{pmatrix} \frac{d\mathbf{v}_\theta(t)}{d\theta}. \quad (5.16)$$

The initial conditions for (5.16) are obtained from the initial conditions for  $\mathbf{p}$  and  $\mathbf{R}$ , that is,  $\mathbf{p}_0(\theta) = \mathbf{p}_0$  and  $\mathbf{R}_0(\theta) = \mathbf{R}_0$ . With these initial conditions, the solution of (5.16) at  $T$  can be represented as

$$\begin{pmatrix} \mathbf{w}_T(\theta) \\ \mathbf{s}_T(\theta) \end{pmatrix} = \int_0^T \Phi_\theta(T, t) \mathbf{B}_\theta(t) \frac{d\mathbf{v}_\theta(t)}{d\theta} dt, \quad (5.17)$$

where the fundamental matrix  $\Phi_\theta(T, t)$  satisfies the evolution equation

$$\frac{\partial \Phi_\theta(t, s)}{\partial t} = \mathbf{A}_\theta(t) \Phi_\theta(t, s), \quad \Phi_\theta(s, s) = \mathbf{I}_6,$$

and

$$\begin{aligned} \mathbf{A}_\theta(t) &= \begin{bmatrix} \mathbf{A}_{11\theta}(t) & \mathbf{A}_{12\theta}(t) \\ \mathbf{0} & -[\mathbf{v}_\theta(t)]_\times \end{bmatrix}, \\ \mathbf{B}_\theta(t) &= \begin{bmatrix} \mathbf{B}_{1\theta}(t) \\ -\mathbf{I} \end{bmatrix}. \end{aligned}$$

The integral operator in (5.17) can be regarded as a Jacobian operator of the 3S<sub>n</sub>PU parallel non-holonomic robot [2]. It has been proved in [1] that the error vanishing formula (5.13) is equivalent to the integral equation

$$\int_0^T \Phi_\theta(T, t) \mathbf{B}_\theta(t) \frac{d\mathbf{v}_\theta(t)}{d\theta} dt = -\gamma \begin{pmatrix} \mathbf{p}_T(\theta) - \mathbf{p}_d \\ \mathbf{r}_T(\theta) \end{pmatrix}, \quad (5.18)$$

where  $[\mathbf{r}_T(\theta)]_\times = \log(\mathbf{R}_T(\theta) \mathbf{R}_d^T)$ . Then, the motion planning algorithm for the studied robot is obtained by solving Eq. (5.18) using a generalized inverse of the Jacobian. If the Moore–Penrose pseudo inverse is chosen, the resulting differential equation for the control function  $\mathbf{v}_\theta(t)$  takes the following form

$$\frac{d\mathbf{v}_\theta(t)}{d\theta} = -\gamma \mathbf{B}_\theta^T(t) \Phi_\theta^T(T, t) \mathbf{D}_\theta^{-1} \begin{pmatrix} \mathbf{p}_T(\theta) - \mathbf{p}_d \\ \mathbf{r}_T(\theta) \end{pmatrix}. \quad (5.19)$$

The matrix  $\mathbf{D}_\theta = \int_0^T \Phi_\theta(T, t) \mathbf{B}_\theta(t) \mathbf{B}_\theta^T(t) \Phi_\theta^T(T, t) dt$  is the Gram matrix of the system (5.16). Given the system (5.19), the solution of the motion planning problem

is computed by taking the limit  $\mathbf{v}(t) = \lim_{\theta \rightarrow +\infty} \mathbf{v}_\theta(t)$ . The system (5.10) subject to the control  $\mathbf{v}(t)$  produces a trajectory  $(\mathbf{p}(t), \mathbf{R}(t))$ . Finally, a suitable substitutions to the feedback Eq. (5.9) defines the control  $\mathbf{u}(t)$  solving the motion planning problem.

## 5.2 Using Truncated Series

Since the motion planning algorithm (5.19) operates in an infinite dimensional space of control functions, its computer implementation needs to be preceded by the introduction of a finite dimensional space of controls. If this is performed by representing the control function  $v_i(t)$ ,  $i = 1, 2, 3$ , by their truncated Fourier series expansion [2], each control can be represented as:

$$v_i(t) = \lambda_{i,0} + \sum_{k=1}^h \left( \lambda_{i,2k-1} \sin \frac{2\pi}{T} kt + \lambda_{i,2k} \cos \frac{2\pi}{T} kt \right). \quad (5.20)$$

As a consequence, the control is finitely parametrized by

$$\Lambda = (\lambda_{1,0}, \dots, \lambda_{1,2h}, \dots, \lambda_{3,0}, \dots, \lambda_{3,2h})^T \in \mathbb{R}^{6h+3}. \quad (5.21)$$

In the finite dimensional case the control family takes the form  $\mathbf{v}_\theta(t) = \mathbf{P}(t)\Lambda(\theta)$ , where the matrix  $\mathbf{P}(t)$  aggregates the basic harmonic functions.

Now, the differential equation (5.19) underlying the motion planning algorithm determines the control coefficients  $\Lambda$

$$\frac{d\Lambda_\theta}{d\theta} = -\gamma \mathbf{J}_{\mathbf{p}_0, \mathbf{R}_0, T}^\#(\Lambda_\theta) \begin{pmatrix} \mathbf{p}_T(\theta) - \mathbf{p}_d \\ \mathbf{r}_T(\theta) \end{pmatrix}, \quad (5.22)$$

where  $\mathbf{J}_{\mathbf{p}_0, \mathbf{R}_0, T}^\#(\Lambda_\theta)$  denotes the Moore–Penrose pseudo inverse of the  $6 \times (6h + 3)$  Jacobian matrix

$$\mathbf{J}_{\mathbf{p}_0, \mathbf{R}_0, T}(\Lambda_\theta) = \int_0^T \Phi_\theta(T, t) \mathbf{B}_\theta(t) \mathbf{P}(t) dt, \quad (5.23)$$

of the  $3S_nPU$  robot.

If the differential equation (5.22) is numerically integrated using the Euler scheme, the following difference equation for  $\Lambda_\theta$ , where  $\theta = 0, 1, \dots$ , is obtained:

$$\Lambda_{\theta+1} = \Lambda_\theta - \gamma \mathbf{J}_{\mathbf{p}_0, \mathbf{R}_0, T}^\#(\Lambda_\theta) \begin{pmatrix} \mathbf{p}_T(\theta) - \mathbf{p}_d \\ \mathbf{r}_T(\theta) \end{pmatrix}. \quad (5.24)$$

### 5.3 Example

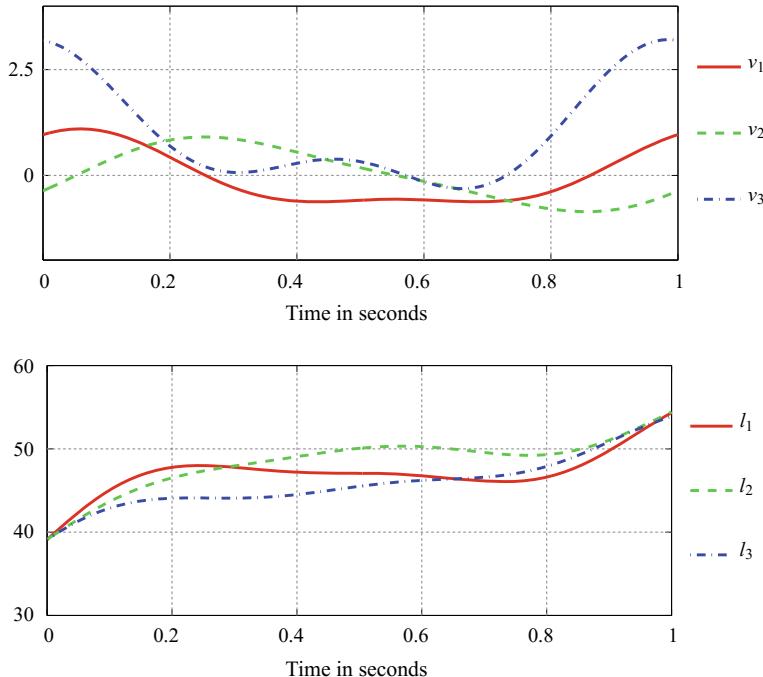
Next, the performance of the presented motion planner is illustrated with a numeric example.

Let us take the same robot used in Sect. 4.6 where the initial position of the platform is  $\mathbf{p}_0 = (0, 0, 25)^T$ , while its orientation  $\mathbf{R}_0 = RPY(0, 0, -\pi/6)$  corresponds to the roll, pitch, and yaw angles 0, 0, and  $-\pi/6$ , respectively. The desired position and orientation for the moving platform  $\mathbf{p}_d = (-0.4, -0.2, 35)^T$  and  $\mathbf{R}_d = RPY(0, 0, -\pi/2)$ .

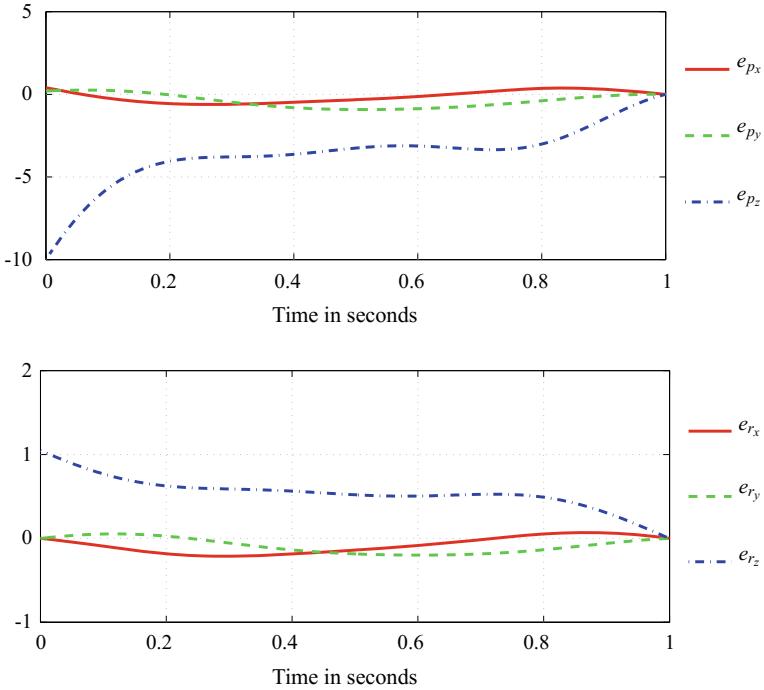
The initial values of control parameters have been set to 0, except for  $\lambda_{11} = \lambda_{21} = \lambda_{32} = 0.5$ ,  $\lambda_{30} = 1$ . The planning time horizon  $T$  is set to 1. The process has been stopped when

$$\mathcal{E}(\theta) = \sqrt{\|\mathbf{p}_T(\theta) - \mathbf{p}_d\|^2 + \|\mathbf{r}_T(\theta)\|^2} < 10^{-3}.$$

In the computations the number  $h$  of harmonics is set to two. The obtained result for control variables are plotted in Fig. 5.1. The evolution of the error in position and orientation are plotted in Fig. 5.2. The error vectors are defined as follows:



**Fig. 5.1** Solution of the motion planning problem for the presented example: controls  $\mathbf{v}(t)$  and leg lengths  $\mathbf{l}(t)$



**Fig. 5.2** Trajectory relative to destination for the presented example: position error  $\mathbf{e}_p = \mathbf{p}(t) - \mathbf{p}_d$  and orientation error  $\mathbf{e}_r(t)$ , where  $[\mathbf{e}_r(t)]_\times = \log(\mathbf{R}(t)\mathbf{R}_d^T)$

$$\mathbf{e}_p(t) = \mathbf{p}(t) - \mathbf{p}_d, \quad (5.25)$$

$$[\mathbf{r}(t)]_\times = \log(\mathbf{R}(t)\mathbf{R}_d^T). \quad (5.26)$$

We have presented a motion planning algorithm based on an application of the endogenous configuration space approach preceded by a feedback transformation of the system (5.7). The presented results provide a novel motion planning algorithm and essentially extend the applicability of the endogenous configuration space approach.

Besides the controllability assumption, the presented algorithm assumes that matrices  $\mathbf{G}$  and  $\mathbf{S}$  are invertible. Geometrically, this means that that vectors  $\mathbf{g}_1$ ,  $\mathbf{g}_2$ ,  $\mathbf{g}_3$  should not be either parallel or co-planar, which is guaranteed by design unless the moving platform of the robot coincides with the base. The fact that matrix  $\mathbf{S}$  can be inverted does not have a so transparent geometric interpretation (see Sect. 4.4).

In principle, it is possible to devise a motion planning algorithm without these assumptions, however, at the expense of increased computational complexity of the algorithm. This trade-off should be subject to further studies.

Since the presented motion planner relies on a Jacobian inversion, only a local convergence can be guaranteed. This means that the initial vector of control parameters should be chosen with some care. This question in the context of parallel

non-holonomic robots has not been addressed before. An issue traditionally raised in the context of the motion planner resulting from (5.19) refers to the choice of basic control functions and of the integration procedure. Alternative choices to those made here could be examined.

## References

1. J. Jakubiak, K. Tchoń, W. Magiera, Motion planning in velocity affine mechanical systems. *Int. J. Control* **83**(9), 1965–1974 (2010)
2. K. Tchoń, J. Jakubiak, Endogenous configuration space approach to mobile manipulators: a derivation and performance assessment of Jacobian inverse kinematics algorithms. *Int. J. Control* **76**(14), 1387–1419 (2003)
3. K. Tchoń, J. Jakubiak, P. Gorsch, F. Thomas, Motion planning for parallel Robots with non-holonomic joints, *Latest Advances in Robot Kinematics* (2012), pp. 115–122

## Chapter 6

# Kinematics of the $S_n$ -2UPS Spherical Robot

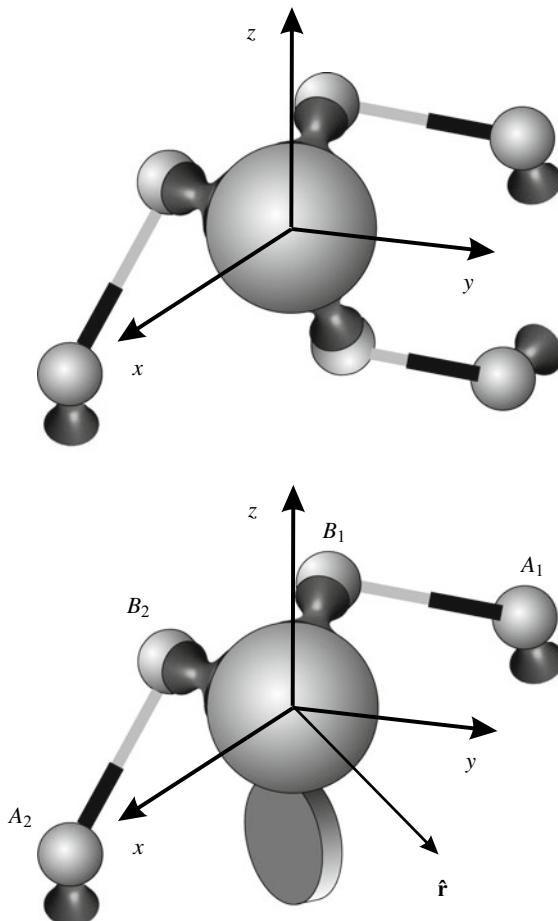


In this chapter, we analize the kinetostatics of a non-holonomic parallel spherical robot which can maneuver to reach any orientation for its moving platform. We show how by properly locating the actuators, and by representing the platform orientation using Euler parameters, the analysis admits a simple bilinear formulation after introducing a local feedback transformation. Interestingly enough, the singularities introduced by this transformation coincide with the singularities of the robot Jacobian. Thus, from the practical point of view, no extra singularities are added. A complete description of the robot's workspace, which also takes into account the limits of all joints, is presented. Part of this work has previously been published in [4].

Consider a S-3SPS robot. This robot consists of a sphere whose center is fixed with respect to the base reference frame and whose orientation is controlled by three prismatic actuators anchored by their ends to the sphere and the base through spherical joints, as shown in Fig. 6.1 (left). This kind of robot has been studied by several authors due to their practical interest as a robotics wrist or, in general, as an orienting device. The works of Innocenti and Parenti-Castelli [7], and Wohlhart [17], are usually referred as the pioneering ones on the kinematics analysis of this parallel robot.

As explained in Chap. 3, it is possible to substitute one of the SPS legs if the S joint directly connecting the base and the moving platform is substituted with a  $S_n$  as shown in Fig. 6.1 (right). This idea was first introduced by Stammers in [14], and generalized by Di Gregorio [3] following the ideas presented by Grosch et al. [5]. The resulting robot can be referred as a  $S_n$ -2UPS robot.

If the system is analyzed at a first-order kinematic level (the dynamics of the system is not considered), it can be shown that the differential equations that describe the system can be expressed in the standard form of two-input driftless (no motion takes place for null inputs) non-holonomic system. If the dynamics of the system is introduced, the system will exhibit drift but an invertible feedback control can eliminate the dynamic parameters [2]. Therefore, the analysis of the system can be



**Fig. 6.1** In a parallel spherical robot actuated by three prismatic joints (top), one of three actuated legs can be eliminated if a disk that rolls without slippage is put in contact with the sphere thus obtaining a non-holonomic spherical joint (bottom)

addressed as that of a two-input driftless non-holonomic system. An important class of non-holonomic systems for which a satisfactory understanding has been reached is the class of systems that can be put, by feedback transformation, in the so-called *chained form* [11]. A complete characterization of such systems (i.e., necessary and sufficient conditions for the existence of a feedback transformation to chained-form) has been provided by [10], while an algorithm for finding the necessary coordinate transform has been presented in [16]. This is important in the presented problem because it has been shown that a two-input driftless non-holonomic system with up to four generalized coordinates can always be transformed in chained form [6, 9, 11]. Once in chained form, different methods can be used for motion planning.

Essentially two kinds of steering inputs signals have been considered: sinusoidal and piecewise constant. While the first approach was pioneered by [11], the second is attributed to [8].

Putting a system in chained form is not an easy task and the result is not always satisfactory. The generated feedbacks introduce, in general, singularities that lead to unfeasible control inputs (i.e., infinite steering rates). Moreover, the characterization of these singularities, in the general case, is difficult due to the complexity of the generated expressions. Besides this, the standard procedures to derive chained forms assume that the number of generalized coordinates coincides with the number of degrees of freedom of the system. This leads to an important drawback when working with spatial orientations as they cannot be parameterized by only three parameters without introducing more singularities.

In this chapter, besides deriving the kinematics model of the robot, it is shown how, by properly arranging the actuators and representing the moving platform orientation using Euler parameters, a bilinear model can be derived and this derivation requires an local feedback whose singularities coincide with the singularities of the robot Jacobian. Thus, no extra singularities are added. Then, it is shown how this bilinear model admits a closed-form formula for the motion planning problem by relying on linear algebra arguments. A general motion planner is derived in Chap. 7.

This chapter is organized as follows. The kinematic model of the general parallel spherical robot with one non-holonomic joint and two prismatic actuators is derived in Sect. 6.1. Next, Sect. 6.2 shows how this model can be expressed in bilinear form by properly arranging the actuators. This bilinear form depends on two matrices whose properties are investigated in Sect. 6.4. The singularities of the system are analyzed in Sect. 6.3. In Sect. 6.5, the robot workspace is analyzed taking into account its singularities and the motion ranges of its joints for a particular example.

## 6.1 Kinematic Model

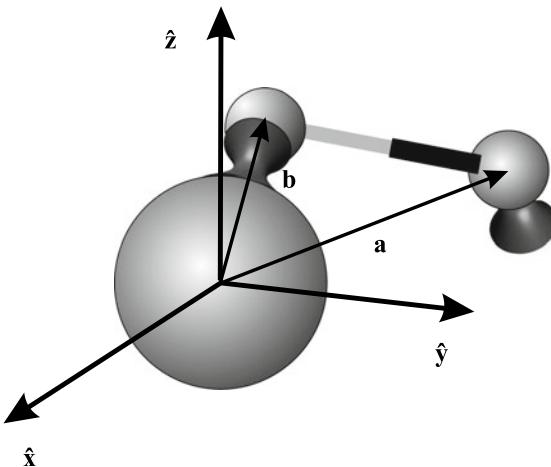
Let us suppose a sphere, centered at the origin, that rotates with angular velocities  $\omega_x$  rad/s,  $\omega_y$  rad/s, and  $\omega_z$  rad/s, about the  $x$ ,  $y$ , and  $z$  axes, respectively. The linear velocity, due to these angular velocities, of a point attached to this sphere with reference position vector  $\mathbf{b}$  is

$$\mathbf{v} = \boldsymbol{\omega} \times \mathbf{b},$$

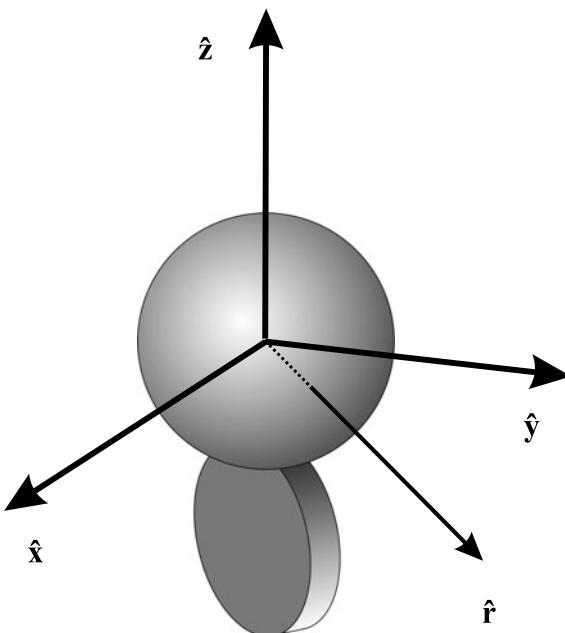
where  $\boldsymbol{\omega} = (\omega_x, \omega_y, \omega_z)^T$ . Then, the linear velocity of this point along the direction given by the unit vector  $\hat{\mathbf{g}}$  is

$$\dot{l} = \hat{\mathbf{g}} \cdot (\boldsymbol{\omega} \times \mathbf{b}) = \boldsymbol{\omega} \cdot (\mathbf{b} \times \hat{\mathbf{g}}). \quad (6.1)$$

Now, if we introduce a prismatic actuator anchored by its ends to the rotating sphere and the world through spherical joints, as depicted in Fig. 6.2, one degree of freedom of the rotating sphere is constrained according to (6.1), where  $\hat{\mathbf{g}}$  is a unit vector in the direction of the actuator and  $\dot{l}$ , its linear velocity.



**Fig. 6.2** A holonomic constraint is imposed on a freely rotating sphere by attaching a prismatic actuator anchored by its ends to the rotating body and the world through spherical joints



**Fig. 6.3** A non-holonomic constraint is imposed on a freely rotating sphere by putting in contact with it a disk that freely rolls without slipping

Alternatively to the constraint introduced above, we can also constrain the motion of the freely rotating sphere by putting in contact with it a disk that rolls without slipping as shown in Fig. 6.3. This disk prevents the sphere to rotate about the axis oriented in the direction of the wheel. In other words,

$$\omega \cdot \hat{\mathbf{r}} = 0. \quad (6.2)$$

Now, let us consider the S-SPS robot in Fig. 6.1 (bottom) in which the rotation of the sphere is constrained by two actuated prismatic joints and a disk. In this case, the angular velocity of the sphere must satisfy the following system of equations

$$\left. \begin{aligned} \dot{l}_1 &= \omega \cdot (\mathbf{b}_1 \times \hat{\mathbf{g}}_1) \\ \dot{l}_2 &= \omega \cdot (\mathbf{b}_2 \times \hat{\mathbf{g}}_2) \\ 0 &= \omega \cdot \hat{\mathbf{r}} \end{aligned} \right\},$$

which can be expressed in matrix form as

$$\mathbf{J}\omega = \begin{pmatrix} \dot{l}_1 \\ \dot{l}_2 \\ 0 \end{pmatrix},$$

where

$$\mathbf{J} = (\mathbf{b}_1 \times \hat{\mathbf{g}}_1 \ \mathbf{b}_2 \times \hat{\mathbf{g}}_2 \ \hat{\mathbf{r}})^T. \quad (6.3)$$

Since  $\hat{\mathbf{g}}_i = (\mathbf{b}_i - \mathbf{a}_i)/l_i$  and  $\mathbf{b}_i = \mathbf{R}\mathbf{b}_i^e$  (where  $\mathbf{R}$  is a  $3 \times 3$  rotation matrix defining the orientation of the moving platform and  $\mathbf{b}_i^e$ , the position vector of leg attachment  $i$  to the moving platform in the moving platform reference frame) the above expression for  $\mathbf{J}$  can be rewritten as:

$$\mathbf{J} = \begin{pmatrix} 1/l_1 & 0 & 0 \\ 0 & 1/l_2 & 0 \\ 0 & 0 & 1 \end{pmatrix} (\mathbf{a}_1 \times \mathbf{R}\mathbf{b}_1^e \ \mathbf{a}_2 \times \mathbf{R}\mathbf{b}_2^e \ \hat{\mathbf{r}})^T. \quad (6.4)$$

Therefore,

$$\omega = \mathbf{K} \begin{pmatrix} \dot{l}_1 l_1 \\ \dot{l}_2 l_2 \end{pmatrix}, \quad (6.5)$$

where

$$\mathbf{K} = \left[ (\mathbf{a}_1 \times \mathbf{R}\mathbf{b}_1^e \ \mathbf{a}_2 \times \mathbf{R}\mathbf{b}_2^e \ \hat{\mathbf{r}})^T \right]^{-1} \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix}. \quad (6.6)$$

## 6.2 Deriving a Bilinear Model

Although three is the minimum number of parameters required to describe the kinematics of a rotating rigid body, every such three-dimensional parametrization of the motion is singular. This is the case of the Euler angles. Alternatively, a non-singular parameterization is possible by using four parameters. This is the case of the Euler parameters defined as

$$\mathbf{q} = \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} = \begin{pmatrix} \cos \frac{\phi}{2} \\ n_x \sin \frac{\phi}{2} \\ n_y \sin \frac{\phi}{2} \\ n_z \sin \frac{\phi}{2} \end{pmatrix}, \quad (6.7)$$

where  $\hat{\mathbf{n}} = (n_x, n_y, n_z)^T$  is the equivalent axis of rotation and  $\phi$ , the angle rotated about it. From this definition, one can easily derive the following constraint

$$\|\mathbf{q}\|^2 = a^2 + b^2 + c^2 + d^2 = 1. \quad (6.8)$$

See [12, 13] for a detailed analysis of Euler parameter and their connections with other parameterizations.

It can be shown that the rotation matrix, in terms of Euler parameters, can be expressed as

$$\mathbf{R} = 2 \begin{pmatrix} a^2 + b^2 - \frac{1}{2} & bc - ad & bd + ac \\ bc + ad & a^2 + c^2 - \frac{1}{2} & cd - ab \\ bd - ac & cd + ab & a^2 + d^2 - \frac{1}{2} \end{pmatrix}. \quad (6.9)$$

If we substitute this parametrization of  $\mathbf{R}$  in (6.6), the result is rather awkward. Nevertheless, an important simplification is attained if the spherical joints of the two legs are oriented at  $\pi/2$  one from each other in their local reference frames. For example, if we set  $\mathbf{a}_1 = (1, 0, 0)^T$ ,  $\mathbf{a}_2 = (0, 1, 0)^T$ ,  $\mathbf{b}_1^e = k\mathbf{a}_1$ , and  $\mathbf{b}_2^e = k\mathbf{a}_2$ , the substitution of (6.9) in (6.6) yields

$$\mathbf{K} = \frac{2k}{\det(\mathbf{J})} \begin{pmatrix} -r_2(ad + bc) + r_3(ac - bd) & r_2(ad - bc) \\ r_1(ad + bc) & r_1(-ad + bc) + r_3(ab + cd) \\ -r_1(ac - bd) & -r_2(ab + cd) \end{pmatrix},$$

where  $\hat{\mathbf{r}} = (r_1, r_2, r_3)^T$ . Further simplifications are still possible by properly locating the disk. For example, if we set  $\hat{\mathbf{r}} = \left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}, 0\right)^T$ , then

$$\mathbf{K} = \frac{\sqrt{2}k}{\det(\mathbf{J})} \begin{pmatrix} -ad - bc & ad - bc \\ ad + bc & bc - ad \\ bd - ac & -ab - cd \end{pmatrix}. \quad (6.10)$$

Since the relationship between angular velocities and time derivatives of Euler parameters is given by

$$\dot{\mathbf{q}} = \frac{1}{2} \begin{pmatrix} -b & -c & -d \\ a & -d & c \\ d & a & -b \\ -c & b & a \end{pmatrix} \omega, \quad (6.11)$$

the substitution of (6.5), with the expression of  $\mathbf{K}$  given in (6.10), in (6.11) yields

$$\dot{\mathbf{q}} = \begin{pmatrix} -b & -c & -d \\ a & -d & c \\ d & a & -b \\ -c & b & a \end{pmatrix} \begin{pmatrix} -ad - bc & ad - bc \\ ad + bc & bc - ad \\ bd - ac & -ab - cd \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix}, \quad (6.12)$$

where

$$u_i = \left( \frac{\sqrt{2}kl_i}{\det(\mathbf{J})} \right) l_i. \quad (6.13)$$

Equation (6.13) can be seen as a transformation in the input variables. It actually represents a local feedback transformation because both  $\det(\mathbf{J})$  and  $l_i$  depend on  $\mathbf{q}$ . Observe that this change of inputs is singular at those orientations in which  $\det(\mathbf{J}) = 0$ . These singularities are studied in Sect. 6.3.

Now, let us define the transformation in the new input variables defined by

$$\begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} ad+bc & bc-ad \\ bd-ac & -ab-cd \end{pmatrix}^{-1} \begin{pmatrix} -\frac{1}{\sqrt{2}} & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}. \quad (6.14)$$

This is also a local feedback transformation because it depends on the orientation of the moving platform. Those orientations for which the matrix inverse in (6.14) is not defined are singularities introduced by this transformation. These singularities are also analyzed in Sect. 6.3 where it is shown that they coincide with the singularities of the robot Jacobian. With this input transformation, (6.12) can be rewritten as

$$\dot{\mathbf{q}} = \begin{pmatrix} -b & -c & -d \\ a & -d & c \\ d & a & -b \\ -c & b & a \end{pmatrix} \begin{pmatrix} -\frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{2}} & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix},$$

or, alternatively, as

$$\dot{\mathbf{q}} = (\mathbf{A}v_1 + \mathbf{B}v_2)\mathbf{q}, \quad (6.15)$$

where

$$\mathbf{A} = \frac{1}{\sqrt{2}} \begin{pmatrix} 0 & 1 & -1 & 0 \\ -1 & 0 & 0 & -1 \\ 1 & 0 & 0 & -1 \\ 0 & 1 & 1 & 0 \end{pmatrix}, \quad (6.16)$$

and

$$\mathbf{B} = \begin{pmatrix} 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}. \quad (6.17)$$

This corresponds to the model of a driftless bilinear system with two inputs and four states, but it is not a minimal representation because the four states are not independent. They must satisfy (6.8). That is,  $\mathbf{q} \in \mathbb{S}^3$  where  $\mathbb{S}^3 = \{\mathbf{x} \in \mathbb{R}^3, \|\mathbf{x}\|^2 = 1\}$ . This dependency is already implicit in (6.15). To make it explicit, let us derive (6.8) with respect to time to obtain

$$\mathbf{q}^T \dot{\mathbf{q}} = 0. \quad (6.18)$$

Then, by substituting (6.15) in (6.18), we have

$$v_1 \mathbf{q}^T \mathbf{A} \mathbf{q} + v_2 \mathbf{q}^T \mathbf{B} \mathbf{q} = 0.$$

Since the above equation must hold for any value of  $v_1$  and  $v_2$ , it can be concluded that

$$\mathbf{q}^T \mathbf{A} \mathbf{q} = 0, \quad (6.19)$$

and

$$\mathbf{q}^T \mathbf{B} \mathbf{q} = 0, \quad (6.20)$$

but the quadratic form of a matrix is identically 0 if, and only if, the matrix is skew-symmetric, as is our case.

### 6.3 Singularities

The singularities of the studied robot are the set of orientations in which  $\det(\mathbf{J}) = 0$ . From (6.3), it can be concluded that they correspond to those orientations in which the vectors  $\mathbf{a}_1 \times \mathbf{b}_1$ ,  $\mathbf{a}_2 \times \mathbf{b}_2$ , and  $\hat{\mathbf{r}}$  lie on a plane. Besides this general case, we have the following particular cases in which  $\det(\mathbf{J}) = 0$ :

1.  $\hat{\mathbf{n}}_1$  or  $\hat{\mathbf{n}}_2$  are not defined. This occurs when  $\mathbf{a}_1$  is aligned with  $\mathbf{b}_1$ , or when  $\mathbf{a}_2$  is aligned with  $\mathbf{b}_2$ . They will always be present, no matter the robot design parameters.
2.  $\hat{\mathbf{n}}_1$  is aligned  $\hat{\mathbf{n}}_2$ . Since  $\mathbf{a}_1$  is not aligned with  $\mathbf{a}_2$  by construction, this only occurs when the base and the platform planes coincide. This singularities always are present no matter the robot design parameters.
3.  $\hat{\mathbf{n}}_1$  is aligned with  $\hat{\mathbf{r}}$ , or  $\hat{\mathbf{n}}_2$  is aligned with  $\hat{\mathbf{r}}$ , or all three vectors are aligned.

For the special configuration with bilinear formulation, the expansion of (6.4) in terms of Euler parameters permits to reformulate  $\det(\mathbf{J}) = 0$  as

$$d^2a(b - c) + c^2b(a - d) - b^2c(a + d) - a^2d(b + c) = 0. \quad (6.21)$$

The substitution of these parameters by their definition given in (6.7) yields

$$(\cos \phi - 1)(p \cos \phi + q \sin \phi + r) = 0,$$

where

$$\begin{aligned} p &= n_x n_z (1 - n_y^2) + n_y n_z (1 - n_x^2), \\ q &= n_y (1 - n_y^2) - n_x (1 - n_x^2), \\ r &= n_x n_z (1 + n_y^2) + n_y n_z (1 + n_x^2). \end{aligned}$$

Then, the configuration is singular if, and only if,  $\phi = 0$ , or

$$\phi = \text{atan2}(q, p) \pm \arccos\left(\frac{-r}{\sqrt{p^2 + q^2}}\right).$$

To derive the bilinear model presented in Sect. 6.2, two input transformations are needed that might introduce extra singularities. The first input transformation (6.13) is only singular in a robot Jacobian singularity, so it does not introduce any new singularity. The second input transformation (6.14) is apparently more complicated but the expansion of the determinant of the matrix that depends on the configuration yields

$$-a^2 d(b+c) - b^2 c(a+c) + c^2 b(d-a) + d^2 a(c-b) = 0,$$

which is identical to (6.21), so it does not introduce any new singularity either.

In Sect. 6.5, the singularities will be revisited in a more general setting.

## 6.4 A, B, and Rotations in $\mathbb{R}^4$

Let us define

$$\mathbf{C} = \mathbf{AB} = \frac{1}{\sqrt{2}} \begin{pmatrix} 0 & 1 & 1 & 0 \\ -1 & 0 & 0 & 1 \\ -1 & 0 & 0 & -1 \\ 0 & -1 & 1 & 0 \end{pmatrix}. \quad (6.22)$$

Then, it can be checked that

$$\mathbf{A}^2 = \mathbf{B}^2 = \mathbf{C}^2 = \mathbf{ABC} = -\mathbf{I}. \quad (6.23)$$

Hamilton called quadruples with these rules of multiplication a *quaternion*. Actually, (6.23) reproduces the celebrated formula that Hamilton carved into the stone of Brougham Bridge. Therefore, the real linear span of  $\{\mathbf{I}, \mathbf{A}, \mathbf{B}, \mathbf{C}\}$  is isomorphic to the real algebra of quaternions. As with standard quaternions, (6.23) determines all the possible products of  $\mathbf{A}$ ,  $\mathbf{B}$ , and  $\mathbf{C}$  resulting in

$$\begin{aligned} \mathbf{AB} &= \mathbf{C}, & \mathbf{BA} &= -\mathbf{C}, \\ \mathbf{BC} &= \mathbf{A}, & \mathbf{CB} &= -\mathbf{A}, \\ \mathbf{CA} &= \mathbf{B}, & \mathbf{AC} &= -\mathbf{B}. \end{aligned} \quad (6.24)$$

According to (6.23), it can be said that  $\mathbf{A}$ ,  $\mathbf{B}$ , and  $\mathbf{C}$  behave as imaginary magnitudes. Then, it is not surprising that their matrix exponentials, defined according to the traditional power series, have simple expressions similar to Euler's formula:

$$e^{\omega\mathbf{A}} = \sin(\omega)\mathbf{A} + \cos(\omega)\mathbf{I}, \quad (6.25)$$

$$e^{\omega\mathbf{B}} = \sin(\omega)\mathbf{B} + \cos(\omega)\mathbf{I}, \quad (6.26)$$

$$e^{\omega\mathbf{C}} = \sin(\omega)\mathbf{C} + \cos(\omega)\mathbf{I}. \quad (6.27)$$

Then, it is not either surprising to realize that  $e^{\omega\mathbf{A}}$ ,  $e^{\omega\mathbf{B}}$  and  $e^{\omega\mathbf{C}}$  behave as rotations in four dimensions. Indeed, since the exponential of an antisymmetric matrix is an orthogonal matrix with determinant equal to +1 and unit length eigenvalues,  $e^{\omega\mathbf{A}}$ ,  $e^{\omega\mathbf{B}}$  and  $e^{\omega\mathbf{C}}$  represent rotations.

According to Cayley's factorization, a 4D rotation matrix can always be expressed as the product of two matrices of the form

$$\mathbf{R}^L(l_1, l_2, l_3, l_4) = \begin{pmatrix} l_1 & -l_2 & -l_3 & -l_4 \\ l_2 & l_1 & -l_4 & l_3 \\ l_3 & l_4 & l_1 & -l_2 \\ l_4 & -l_3 & l_2 & l_1 \end{pmatrix}, \quad (6.28)$$

and

$$\mathbf{R}^R(r_1, r_2, r_3, r_4) = \begin{pmatrix} r_1 & -r_2 & -r_3 & -r_4 \\ r_2 & r_1 & r_4 & -r_3 \\ r_3 & -r_4 & r_1 & r_2 \\ r_4 & r_3 & -r_2 & r_1 \end{pmatrix}, \quad (6.29)$$

which are known as left- and right-isoclinic rotation matrices, respectively (see [15] for details on Cayley's factorization).

Now, it can be observed that

$$\mathbf{R}^R(r_1, r_2, r_3, r_4) = \gamma_1\mathbf{I} + \gamma_2\mathbf{A} + \gamma_3\mathbf{B} + \gamma_4\mathbf{C}, \quad (6.30)$$

where

$$\begin{pmatrix} \gamma_1 \\ \gamma_2 \\ \gamma_3 \\ \gamma_4 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 \end{pmatrix} \begin{pmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \end{pmatrix}. \quad (6.31)$$

Hence,

$$\begin{pmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -\frac{\sqrt{2}}{2} & 0 & -\frac{\sqrt{2}}{2} \\ 0 & \frac{\sqrt{2}}{2} & 0 & -\frac{\sqrt{2}}{2} \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} \gamma_1 \\ \gamma_2 \\ \gamma_3 \\ \gamma_4 \end{pmatrix}. \quad (6.32)$$

Therefore,  $\{\mathbf{I}, \mathbf{A}, \mathbf{B}, \mathbf{C}\}$  is a basis for right-isoclinic rotations and, as a consequence, (6.25)–(6.27) represent right-isoclinic rotations.

In what follows, it is shown that three variable consecutive rotations represented by  $\mathbf{A}$ ,  $\mathbf{B}$ , and  $\mathbf{A}$ , covers all the right isoclinic rotational space in  $\mathbb{R}^4$ . This is the key result used in the next chapter to solve the motion planning problem. From (6.25) and (6.26), we obtain

$$\begin{aligned} e^{\omega_3 \mathbf{A}} e^{\omega_2 \mathbf{B}} e^{\omega_1 \mathbf{A}} &= \\ &(\sin(\omega_3)\mathbf{A} + \cos(\omega_3)\mathbf{I})(\sin(\omega_2)\mathbf{B} + \cos(\omega_2)\mathbf{I})(\sin(\omega_1)\mathbf{A} + \cos(\omega_1)\mathbf{I}) \\ &= \sin(\omega_3)\mathbf{A} \sin(\omega_2)\mathbf{B} \sin(\omega_1)\mathbf{A} + \cos(\omega_3)\mathbf{I} \sin(\omega_2)\mathbf{B} \sin(\omega_1)\mathbf{A} \\ &\quad + \sin(\omega_3)\mathbf{A} \cos(\omega_2)\mathbf{I} \sin(\omega_1)\mathbf{A} + \cos(\omega_3)\mathbf{I} \cos(\omega_2)\mathbf{I} \sin(\omega_1)\mathbf{A} \\ &\quad + \sin(\omega_3)\mathbf{A} \sin(\omega_2)\mathbf{B} \cos(\omega_1)\mathbf{I} + \cos(\omega_3)\mathbf{I} \sin(\omega_2)\mathbf{B} \cos(\omega_1)\mathbf{I} \\ &\quad + \sin(\omega_3)\mathbf{A} \cos(\omega_2)\mathbf{I} \cos(\omega_1)\mathbf{I} + \cos(\omega_3)\mathbf{I} \cos(\omega_2)\mathbf{I} \cos(\omega_1)\mathbf{I}. \end{aligned}$$

Then, using Eq. (6.24), we obtain

$$\begin{aligned} e^{\omega_3 \mathbf{A}} e^{\omega_2 \mathbf{B}} e^{\omega_1 \mathbf{A}} &= \sin(\omega_3) \sin(\omega_2) \sin(\omega_1)\mathbf{B} - \cos(\omega_3) \sin(\omega_2) \sin(\omega_1)\mathbf{C} \\ &\quad - \sin(\omega_3) \cos(\omega_2) \sin(\omega_1)\mathbf{I} + \cos(\omega_3) \cos(\omega_2) \sin(\omega_1)\mathbf{A} \\ &\quad + \sin(\omega_3) \sin(\omega_2) \cos(\omega_1)\mathbf{C} + \cos(\omega_3) \sin(\omega_2) \cos(\omega_1)\mathbf{B} \\ &\quad + \sin(\omega_3) \cos(\omega_2) \cos(\omega_1)\mathbf{A} + \cos(\omega_3) \cos(\omega_2) \cos(\omega_1)\mathbf{I}. \end{aligned}$$

That is,

$$\begin{aligned} e^{\omega_3 \mathbf{A}} e^{\omega_2 \mathbf{B}} e^{\omega_1 \mathbf{A}} &= (\cos(\omega_3) \cos(\omega_1) - \sin(\omega_3) \sin(\omega_1)) \cos(\omega_2)\mathbf{I} \\ &\quad + (\cos(\omega_3) \sin(\omega_1) + \sin(\omega_3) \cos(\omega_1)) \cos(\omega_2)\mathbf{A} \\ &\quad + (\sin(\omega_3) \sin(\omega_1) + \cos(\omega_3) \cos(\omega_1)) \sin(\omega_2)\mathbf{B} \\ &\quad + (\sin(\omega_3) \cos(\omega_1) - \cos(\omega_3) \sin(\omega_1)) \sin(\omega_2)\mathbf{C}. \end{aligned}$$

Now, after somewhat tedious algebraic manipulations, it can be checked that:

$$\begin{aligned} e^{\omega_3 \mathbf{A}} e^{\omega_2 \mathbf{B}} e^{\omega_1 \mathbf{A}} &= \cos(\omega_2) \cos(\omega_3 + \omega_1)\mathbf{I} \\ &\quad + \cos(\omega_2) \sin(\omega_3 + \omega_1)\mathbf{A} \\ &\quad + \sin(\omega_2) \cos(\omega_3 - \omega_1)\mathbf{B} \\ &\quad + \sin(\omega_2) \sin(\omega_3 - \omega_1)\mathbf{C}. \end{aligned} \quad (6.33)$$

Therefore, any arbitrary right-isoclinic rotation can be expressed as:

$$\gamma_1 \mathbf{I} + \gamma_2 \mathbf{A} + \gamma_3 \mathbf{B} + \gamma_4 \mathbf{C} = e^{\omega_3 \mathbf{A}} e^{\omega_2 \mathbf{B}} e^{\omega_1 \mathbf{A}}. \quad (6.34)$$

Then, to find an expression for  $\omega_1$ ,  $\omega_2$ , and  $\omega_3$ , we can identify (6.33) and (6.34)

$$\begin{aligned}\gamma_1 &= \cos(\omega_2) \cos(\omega_3 + \omega_1), \\ \gamma_2 &= \cos(\omega_2) \sin(\omega_3 + \omega_1), \\ \gamma_3 &= \sin(\omega_2) \cos(\omega_3 - \omega_1), \\ \gamma_4 &= \sin(\omega_2) \sin(\omega_3 - \omega_1).\end{aligned} \quad (6.35)$$

Therefore, from the first and the second equation above, we have that

$$\omega_3 + \omega_1 = \text{atan2}(\gamma_2, \gamma_1), \quad (6.36)$$

and, from the third and the fourth,

$$\omega_3 - \omega_1 = \text{atan2}(\gamma_4, \gamma_3). \quad (6.37)$$

As a consequence, the subtraction of (6.36) from (6.37) yields

$$\omega_1 = \frac{1}{2}(\text{atan2}(\gamma_2, \gamma_1) - \text{atan2}(\gamma_4, \gamma_3)). \quad (6.38)$$

Likewise, the addition of (6.36) and (6.37) yields

$$\omega_3 = \frac{1}{2}(\text{atan2}(\gamma_2, \gamma_1) + \text{atan2}(\gamma_4, \gamma_3)). \quad (6.39)$$

Now, dividing the third equation by the first one in (6.35), we obtain:

$$\frac{\gamma_3}{\gamma_1} = \frac{\sin(\omega_2) \cos(\omega_3 - \omega_1)}{\cos(\omega_2) \cos(\omega_3 + \omega_1)}.$$

That is,

$$\tan(\omega_2) = \frac{\gamma_3}{\gamma_1} \frac{\cos(\omega_3 + \omega_1)}{\cos(\omega_3 - \omega_1)}.$$

Then, replacing (6.36) and (6.37) in the above expression, we finally obtain:

$$\omega_2 = \arctan \left( \frac{\gamma_3 \cos(\text{atan2}(\gamma_2, \gamma_1))}{\gamma_1 \cos(\text{atan2}(\gamma_4, \gamma_3))} \right). \quad (6.40)$$

The values of  $\omega_1$ ,  $\omega_2$  and  $\omega_3$  obtained using (6.38), (6.40), and (6.39), respectively, permit to solve Eq. (6.33) in closed form.

## 6.5 Workspace Computation

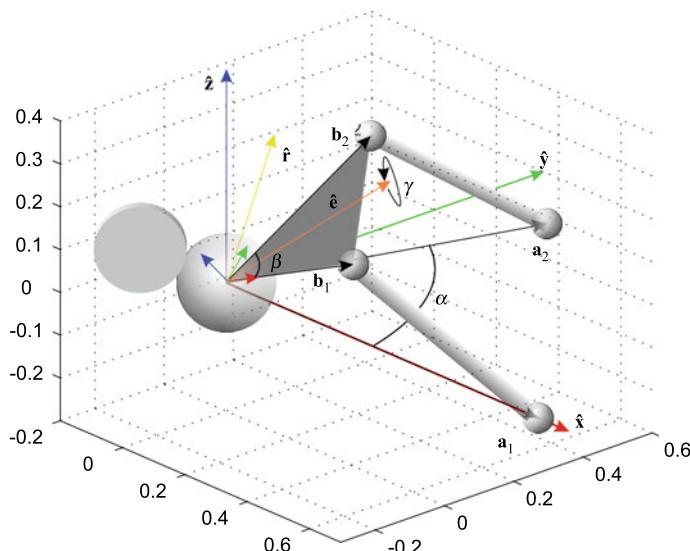
A fundamental element in the design of any robot is the computation of its workspace. In what follows, the workspace of the robot is determined by taking into account separately:

- (1) the singularities of the Jacobian;
- (2) the motion ranges of the prismatic actuators;
- (3) the working range of the S and U joints; and
- (4) the collisions between base, legs and moving platform.

To define the robot design, without loss of generality,  $\mathbf{a}_1$  is set to  $(1, 0, 0)$  and  $\mathbf{a}_2$  is assumed to lie in the  $xy$ -plane of the base reference frame,  $\mathbf{b}_1^e$  will lie on the  $x$ -axis of the moving platform reference frame, and  $\mathbf{b}_2^e$  will lie in the  $xy$ -plane of the latter reference frame. Moreover,  $\alpha$  will denote the angle between  $\mathbf{a}_1$  and  $\mathbf{a}_2$  (i.e.,  $\mathbf{a}_2 = \mathbf{R}_z(\alpha)\mathbf{a}_1$ );  $\beta$ , the angle between  $\mathbf{b}_1^e$  and  $\mathbf{b}_2^e$  (i.e.,  $\mathbf{b}_2^e = \mathbf{R}_z(\beta)\mathbf{b}_1^e$ ); and  $k$ , the scale factor between  $\mathbf{a}_1$  and  $\mathbf{b}_1^e$  (i.e.,  $\mathbf{b}_1^e = k\mathbf{a}_1$ ). As a consequence, the robot will be geometrically defined by  $\alpha$ ,  $\beta$ ,  $k$ , and  $\hat{\mathbf{r}}$  (Fig. 6.4).

### 6.5.1 Graphical Representation of the Platform Orientation

In order to analyze the robot workspace, a parametrization leading to a representation that can be easily and intuitively interpreted, is needed. There are multiple sets of



**Fig. 6.4** Parameters used to geometrically define a  $S_n$ -2SUS robot

parameters to represent the orientation of an object (Euler angles, Euler parameters, vector-angle, etc.) and each of them has its pros and cons. Choosing a point in a plot using any of these sets of parameters and trying to imagine how the platform is oriented is, in general, far from intuitive. The situation worsens when trying to imagine how the orientation changes when following a trajectory in the same plot. With these difficulties in mind, an ad-hoc representation of the platform orientation have been adopted for graphical purposes.

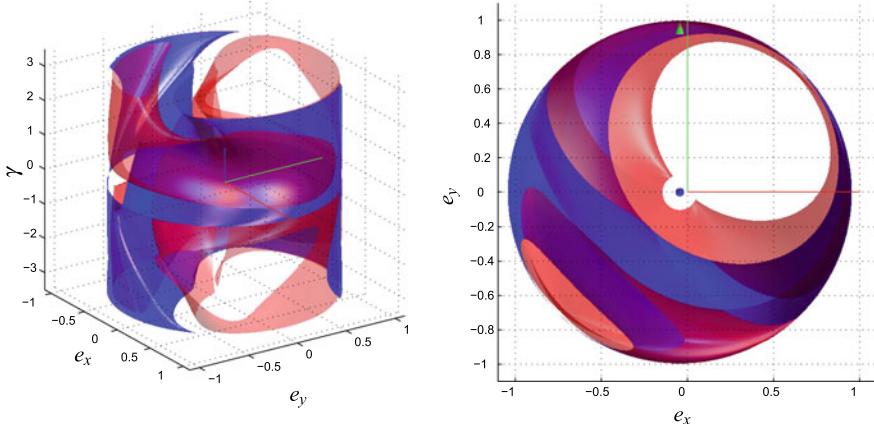
The platform orientation will be represented by a unit vector  $\hat{\mathbf{e}}$  and an angle  $\gamma$  (see Fig. 6.4).  $\hat{\mathbf{e}}$  will be the unit vector pointing from the origin to the midpoint of the segment connecting  $\mathbf{b}_1$  and  $\mathbf{b}_2$ . Since  $\hat{\mathbf{e}}$  is a unit vector, we only need  $e_x$  and  $e_y$  to define it, all possible values for  $e_x$  and  $e_y$  will be inside the unit circle of the  $xy$ -plane. Then, we can represent the platform orientation as the position of the center line of the platform,  $\hat{\mathbf{e}}$ , and the angle  $\gamma$  rotated about it, using the right hand convention. For example, when  $\gamma = 0$ , the line defined by  $\mathbf{b}_1$  and  $\mathbf{b}_2$  is parallel to the  $xy$ -plane. Thus, the scalars  $e_x$ ,  $e_y$  and  $\gamma$  can be used to unambiguously represent the platform orientation. Like any other set of three parameters, this orientation representation has singularities. They correspond to the cases in which  $\hat{\mathbf{e}} = (0, 0, 1)^T$ . When this occurs, any value of  $\gamma$  makes the line defined by  $\mathbf{b}_1$  and  $\mathbf{b}_2$  to be parallel to the  $xy$ -plane, so the orientation of the platform cannot be defined. This corresponds to the case in which the moving platform plane is orthogonal to the base plane. This case excluded from the analysis given below.

### 6.5.2 Workspace Boundaries Due to Singularities

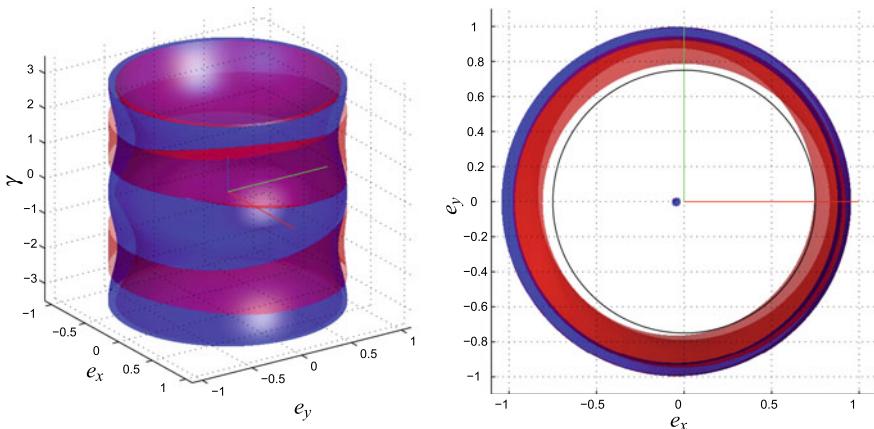
If the robot is near to a singularity, it will still withstand any external force, at least in theory, but the internal forces generated to counteract the external forces would be as high as leading to a possible slippage between the roller and the sphere, or to the break down of some mechanical elements. Thus, in practice, not only the orientations at which  $\det(\mathbf{J}) = 0$  have to be determined, but it is also important to examine the condition number of  $\mathbf{J}$  in the whole workspace. In general, a linear system is said to be ill-conditioned if the logarithm of its defining matrix condition number is higher than the precision of the matrix entries [1]. Thus, in our case, we will say, as a rule of thumb, that an orientation with condition number lower than 10 is well-conditioned.

As an example, let us set the robot design parameters to  $k = 0.5$ ,  $\alpha = \pi/2$ ,  $\beta = \pi/4$  and  $\hat{\mathbf{r}} = (1/\sqrt{3}, 1/\sqrt{3}, 1/\sqrt{3})^T$ . Observe that the kinematics model of this robot is not necessarily bilinear as  $\beta \neq \pi/2$ , but we prefer to analyze a more general case in this section. For this robot, Fig. 6.5 shows two views of the singularity locus, represented as a blue surface. In the same figure, the red surface represents those orientations in which the condition number of  $\mathbf{J}$  is 10. Any orientation delimited by the red and blue surfaces should be avoided, as they are not well-conditioned.

If no other particular objective than maximizing the workspace is considered in the design phase, the best solution is setting  $\hat{\mathbf{r}} = (0, 0, 1)^T$ . In this case, all singularities are limited to the case in which  $\mathbf{b}_1$ , or  $\mathbf{b}_2$ , are coplanar with  $\mathbf{a}_1$  and  $\mathbf{a}_2$ ,

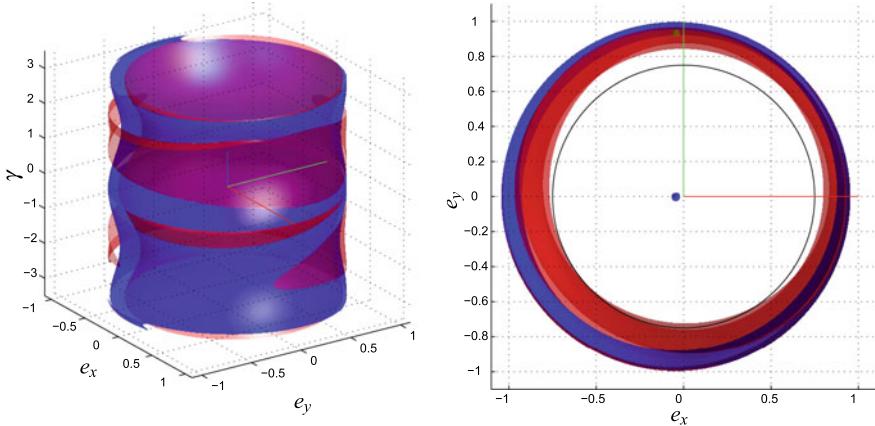


**Fig. 6.5** Singularity analysis of an  $S_n$ -2UPS robot with design parameters  $k = 0.5$ ,  $\alpha = \pi/2$ ,  $\beta = \pi/4$  and  $\hat{\mathbf{r}} = (1/\sqrt{3}, 1/\sqrt{3}, 1/\sqrt{3})^T$ . The blue surfaces represent the singularity locus, and the red surfaces the orientations with condition number equal to 10. Left: Isometric view. Right: Top view



**Fig. 6.6** Singularity analysis of an  $S_n$ -2UPS robot with same design parameters as those in Fig. 6.5 except for  $\hat{\mathbf{r}} = (0, 0, 1)^T$ . The blue surfaces represent the singularity locus, and the red surfaces the orientations with condition number equal to 10. Left: Isometric view. Right: Top view

which makes  $\hat{\mathbf{n}}_1$  aligned with  $\hat{\mathbf{r}}$  or/and  $\hat{\mathbf{n}}_2$  with  $\hat{\mathbf{r}}$ . To see the effect on the singularity locus, let us continue with the previous example, changing only  $\hat{\mathbf{r}} = (0, 0, 1)^T$ . For this particular design, Fig. 6.6 shows two views of the resulting singularity locus as a blue surface, and those configurations in which the condition number is 10 as a red surface. Observe how the region in which  $e_z > \sqrt{1 - 0.8^2}$  can be considered as well-conditioned configurations.

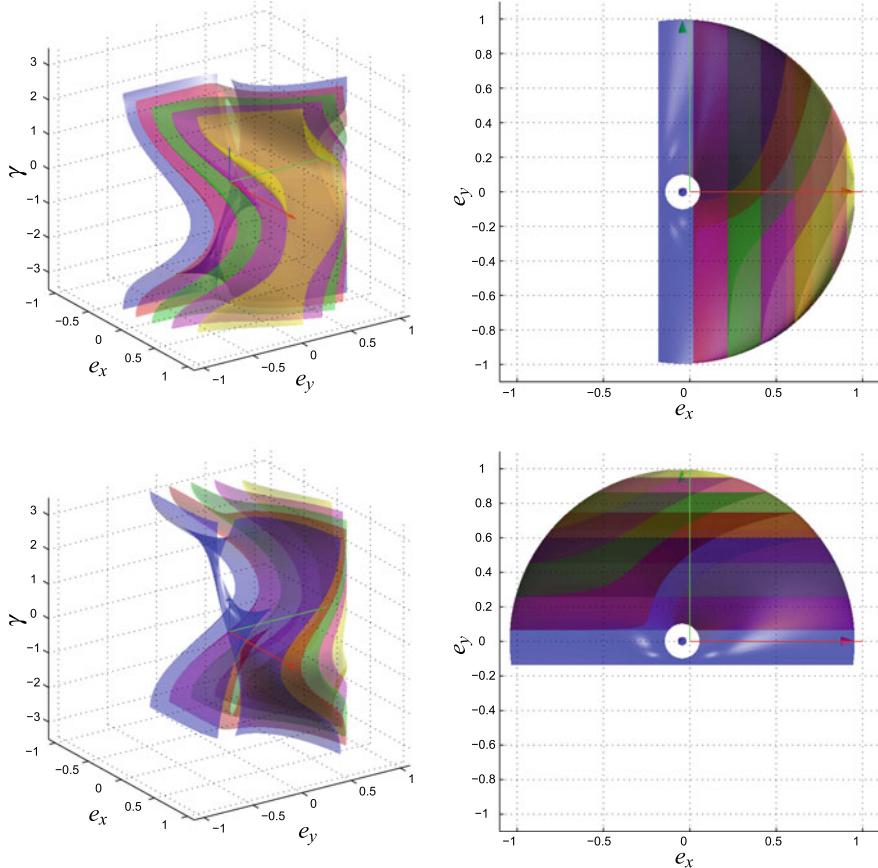


**Fig. 6.7** Singularity analysis of an  $S_n$ -2UPS robot with same design parameters as those in Fig. 6.5 except for  $\hat{\mathbf{r}} = (0.1, 0.1, \sqrt{0.6})^T$ . The blue surfaces represent the singularity locus, and the red surfaces the orientations with condition number equal to 10. Left: Isometric view. Right: Top view

Another interesting aspect to study in the singularity locus is the influence of small variations of  $\hat{\mathbf{r}}$  that could come from construction or assembly inaccuracies. Performing the same singularity locus exploration as to the previous example, but with  $\hat{\mathbf{r}} = (0.1, 0.1, \sqrt{0.6})^T$ , shows an insignificant variation with respect to  $\hat{\mathbf{r}} = (0, 0, 1)^T$ . The resulting singularity locus appears in Fig. 6.7. Thus, it can be concluded that, in general, the choice of  $\hat{\mathbf{r}} = (0, 0, 1)^T$  is a good and robust solution.

### 6.5.3 Workspace Boundaries Due to Joint Limits

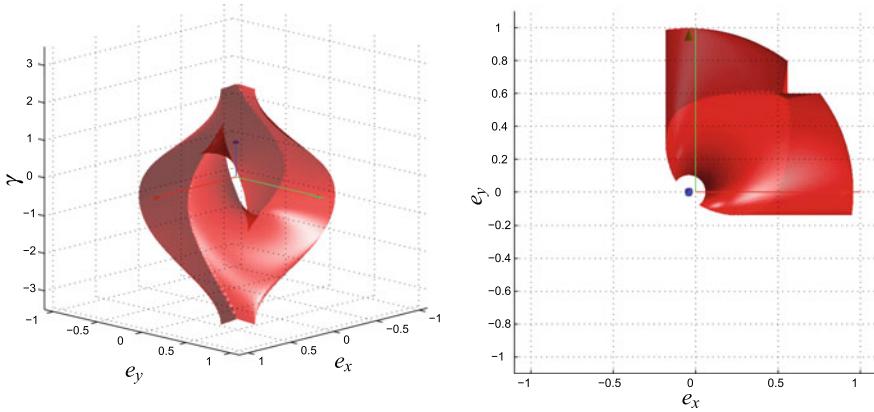
As already pointed out, the workspace is not only determined by the robot singularities. The motion ranges of the prismatic actuators and the working ranges of the spherical and universal joints have to be taken into account for its complete characterization. The workspace due to the motion ranges of the legs in the same example above can be performed by characterizing the platform orientations in a cylindrical mesh and finding the leg length at each discretized orientation. Surfaces are constructed for orientations with equal leg lengths ranging from 1 to 0.5 for each leg. Fig. 6.8 shows four views of the surfaces, two views for each leg. The plot information on where the platform is capable of arriving with each leg, assuming that the other leg has no length restriction. Each plot shows six surfaces. Each surface represent an specific leg length. In the case in which the leg length for leg 1 is 0.5 the surface degenerates into a curve passing through  $e_x = 0.9239$ ,  $e_y = 0.3827$ , and  $\gamma = 0$ .



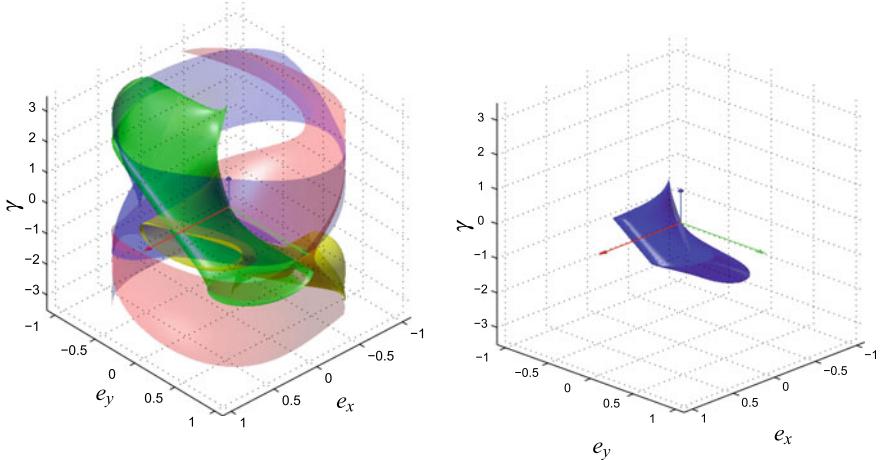
**Fig. 6.8** Contribution of the prismatic joints motion ranges to the analyzed workspace. Each surface corresponds to orientations in which the robot has a leg with a fixed length. The color code is as follows: Leg length: 1 (blue), 0.9 (red), 0.8 (green), 0.7 (magenta) and 0.6 (yellow). Top row: limits due to leg 1. Bottom row: limits due to leg 2. Isometric and zenithal views are shown in both cases

The plots in Fig. 6.8 are obtained assuming that only one leg length is constrained. Thus, the actual robot workspace is the intersection of the contribution of each leg. Figure 6.9 shows the result.

Another factor that limits the workspace is the motion ranges of the universal and spherical joints. Typically, universal joints have a working range of  $35^\circ$  measure from their center line. In other words, they can move inside a  $70^\circ$  cone. Spherical joints have an equivalent work range although they have an additional degree of freedom. They freely spin around the axis of the prismatic actuator. Thus, for our analysis, they can be treated as universal joints. Next, an orientation is chosen to be the reference one. At this orientation all the joints are centered at the middle of their working range. Repeating the same procedure as in the case of leg ranges, the



**Fig. 6.9** Isometric and zenithal views of the robot workspace limits due to the motion ranges of the two prismatic joints. Both ranges are  $[0.5, 1]$ . Left: Isometric view. Right: Top view



**Fig. 6.10** Robot workspace limits due to the motion ranges of the four spherical joints. These ranges are assumed to be cones with  $70^\circ$  of aperture. Left: the four regions resulting from computing the limit for each joint independently. Right: Intersection of all four regions

volumes containing feasible orientations for each joint are found, where the surface of the volume are orientations at the limit of the joint range. The feasibility of an orientation is determined by doing the dot product of the vector along the leg at the actual orientation and the vector along the leg at the reference orientation. The process is done for the four joints [Fig. 6.10 (left)], and then an intersection of the four volumes is found [Fig. 6.10 (right)]. The obtained volume contains all the feasible orientations of the  $S_n$ -2UPS robot, without violating the joints ranges.

Finally, we should find the limits of the workspace due to collisions between all the mechanical elements, but in this particular case the limits due to joint ranges are

so restrictive that there is no possible collision inside the allowed region. Thus, it is not necessary to perform this computation in this case. Moreover, it is interesting to observe that the limits of the workspace due to the joint ranges are more restrictive than those due to the motion ranges of the prismatic actuators, and also to those due to the robot Jacobian singularities.

We have shown how designing a parallel orienting platform with only two actuators becomes feasible by introducing mechanical elements that lead to non-holonomic constraints. The advantages of the presented design might seem dubious when facing the necessity of introducing a path planner to generate the required maneuvers to reach a target from a given initial configuration. Nevertheless, in Chap. 7 we present a general motion planner based on elementary geometric arguments, which overcomes the limitations of the one presented in this chapter, whose simplicity makes the proposed robot architecture a real alternative in some applications where the reduction of bulk, weight or cost is a must.

## References

1. J. Angeles, *Fundamentals of Robotic Mechanical Systems: Theory, Methods, and Algorithms*, Mechanical Engineering Series (2003)
2. A. De Luca, G. Oriolo, Modelling and control of nonholonomic mechanical systems, in *CISM International Centre for Mechanical Sciences (Chapter 7)*, vol. 360 (1995), pp. 277–342
3. R. Di Gregorio, Position analysis and path planning of the S-(nS)PU-SPU and S-(nS)PU-2SPU underactuated wrists. *J. Mech. Robot.* **4**(2) (2012)
4. P. Grosch, F. Thomas, A bilinear formulation for the motion planning of non-holonomic parallel orienting platforms, in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Tokyo, Japan (2013), pp. 953–958
5. P. Grosch, R. Di Gregorio, F. Thomas, Generation of under-actuated manipulators with non-holonomic articulations from ordinary manipulators. *J. Mech. Robot.* **2**(1), 11005–11012 (2010)
6. H. Hermes, Distributions and the Lie algebras their bases can generate. *Proc. Am. Math. Soc.* **106**(2), 555–565 (1989)
7. C. Innocenti, V. Parenti-Castelli, Echelon form solution of direct kinematics for the general fully-parallel spherical wrist. *Mech. Mach. Theory* **28**(4), 553–561 (1993)
8. G. Lafferriere, A general strategy for computing steering controls of systems without drift, in *IEEE Conference on Decision and Control*, Brighton, UK (1991), pp. 1115–1120
9. R.M. Murray, Control of nonholonomic systems using chained forms. *Fields Inst. Commun. Dyn. Control Mech. Syst.* **1**, 219–245 (1993)
10. R.M. Murray, Nilpotent bases for a class of nonintegrable distributions with applications to trajectory generation for nonholonomic systems. *Math. Control Signals Syst.* **7**(1), 58–75 (1994)
11. R.M. Murray, S.S. Sastry, Nonholonomic motion planning: steering using sinusoids. *IEEE Trans. Autom. Control* **38**(5), 700–716 (1993)
12. P.E. Nikravesh, O.K. Kwon, R.A. Wehage, Euler parameters in computational kinematics and dynamics, Part 2. *ASME J. Mech. Transm. Autom. Des.* **107**(3), 366–369 (1985)
13. P.E. Nikravesh, O.K. Kwon, R.A. Wehage, Euler Parameters in Computational Kinematics and Dynamics, Part 1. *ASME J. Mech. Transm. Autom. Des.* **107**(3), 358–365 (1985)
14. C.W. Stammers, P.H. Prest, C.G. Mobley, A friction drive robot wrist: electronic and control requirements. *Mechatronics* **2**(4), 391–401 (1992)
15. F. Thomas, Approaching dual quaternions from matrix algebra. *IEEE Trans. Robot.* **30**(5), 1037–1048 (2014)

16. D. Tilbury, R.M. Murray, S.S. Sastry, Trajectory generation for the n-trailer problem, using Goursat normal form. *IEEE Trans. Autom. Control* **40**(5), 802–819 (1995)
17. K. Wohlhart, Displacement analysis of the general spherical Stewart platform. *Mech. Mach. Theory* **29**(4), 581–589 (1994)

# Chapter 7

## Motion Planning for the $S_n$ -2UPS Robot



As explained in Chap. 6, any system with two inputs and up to four generalized coordinates can always be transformed into chained form. Then, since the  $S_n$ -2UPS non-holonomic robot has two inputs (its leg lengths) and three generalized coordinates (its orientation parameters), its kinematics can be formulated in chained form. Given a system in chain form, its motion planning problem can be solved using well-established procedures (see [7] and the references therein) which means the motion planning problem for the analyzed spherical robot can be solved using one of these procedures, as Jakubiak did in [6] for a particular configuration of the system. In this case, Jakubiak used control functions given by truncated trigonometric series, as we already did in Chap. 5 for the  $3S_n$ PU robot. Nevertheless, the use of these procedures requires a good understanding of sophisticated methods in non-linear control whose technicalities have proven a challenge to many practitioners who are not familiar with them. As an alternative, geometric motion planners have been proposed, for example, in [1]. The main advantage of this kind of motion planners is that they are based on elementary kinematics arguments. However, they generate stepped maneuvers, that is, maneuvers with intermediate instants with zero velocity that guide the moving platform to the desired orientation. This chapter presents a geometric motion planner able to steer the robot to the desired orientation through a differentiable path in the space of actuation variables. Part of the work presented in this chapter appeared in [5].

The motion planning problem for the studied robot can be decomposed into the following two steps: (1) first solve the planning problem considering only the sphere and the disk that constrains its motion, and then (2) obtain the required motion for the prismatic joints in the legs using the inverse kinematics of the robot. Separating both problems, instead of considering both at once, leads to an important simplification. However, this does not go without a price as the first step does not take into account the robot singularities. This is not an important drawback if a workspace free from singularities is defined beforehand.

Based on a rather old result on linear time-varying systems, this chapter shows that there are infinitely many differentiable paths connecting two arbitrary configurations in  $SO(3)$  such that the instantaneous axis of rotation along the path rest on a fixed plane. This theoretical result leads to a practical motion planner for non-holonomic parallel spherical robots. A motion planner which is a closed-form solution and whose derivation requires no other tools than ordinary linear algebra. To present this result, we start with a motion planner based on three moves, and then we proceed by progressively reducing the number of moves to one, thus providing a unified treatment with respect to previous geometric motion planners.

This chapter is organized as follows. In the next section, we briefly summarize the part of kinematic model derived in the previous chapter that is needed by the motion planners presented in the following sections. Then, in Sect. 7.2, we start with a motion planner that generates three-move maneuvers consisting of three consecutive rotations around two orthogonal axes on the plane defined by the non-holonomic constraint. In Sect. 7.3, a two-move maneuver is derived consisting of two consecutive rotations around two non-necessarily orthogonal axes of this plane. Finally, in Sect. 7.4, it is shown how a single rotation around a variable axis that rests on the plane solves the motion planning problem. To compare the procedures, a detailed example is developed in Sect. 7.5. A description of the experimental testbed, where the derived motion planner have been verified, is presented in Sect. 7.6.

## 7.1 Kinematic Model

As we have already said, the motion planning problem for the studied non-holonomic parallel robot can be decomposed into that resulting of considering only the sphere and the disk and that of obtaining the corresponding motion for the prismatic joints using the inverse kinematics of the robot. Next, we briefly summarize the mathematics behind both steps.

Once the disk is put in contact with the sphere, the sphere can still rotate about any axis orthogonal to an axis defined by the contact with the disk. Without loss of generality, we can assume that this constrained axis, say  $\hat{\mathbf{r}}$ , coincides with the  $z$ -axis. Therefore, the sphere can undergo any rotation of the form

$$\mathbf{R}_{\mathbf{n}}(\omega) = \mathbf{R}_z(\theta)\mathbf{R}_x(\omega)\mathbf{R}_z(-\theta), \quad (7.1)$$

where  $\mathbf{n} = [\cos \theta, \sin \theta, 0]^T$ . In particular, it can be rotated about the  $x$ -axis ( $\theta = 0$ ) or the  $y$ -axis ( $\theta = \pi/2$ ).

From the instantaneous kinematics point of view, the sphere can undergo any angular velocity of the form  $\boldsymbol{\omega} = (w_x(t), w_y(t), 0)$ . Therefore, the sphere's orientation must satisfy the differential equation

$$\dot{\mathbf{R}}(t) = \mathbf{W}(t)\mathbf{R}(t), \quad (7.2)$$

where

$$\mathbf{W}(t) = \begin{pmatrix} 0 & 0 & w_y(t) \\ 0 & 0 & -w_x(t) \\ -w_y(t) & w_x(t) & 0 \end{pmatrix}. \quad (7.3)$$

Finding  $\mathbf{R}(t)$  means integrating (7.2). Although this equation describes a non-holonomic dynamic system and hence it is not integrable, it does not mean that, for particular angular velocities, it can be integrated. For example, for the trivial case in which the angular velocities are constant,  $\mathbf{W}$  becomes constant and then, using Rodrigues' formula, we have that

$$\mathbf{R}(t) = e^{\mathbf{W}t} = \mathbf{I} + \frac{\sin(\theta)}{\theta} \mathbf{W} + \frac{1 - \cos(\theta)}{\theta^2} \mathbf{W}^2, \quad (7.4)$$

where  $\theta = t\sqrt{w_x^2 + w_y^2}$ .

The lengths of the prismatic actuators for a given orientation of the sphere,  $\mathbf{R}$ , will be given by

$$l_i = \|\mathbf{b}_i - \mathbf{a}_i\| = \|\mathbf{R}\mathbf{b}_i^0 - \mathbf{a}_i\|, \quad i = 1, 2, \quad (7.5)$$

where  $\mathbf{a}_i$  stands for the location of the center of the  $i$  spherical joint attached to the base,  $\mathbf{b}_i$  for the location of the corresponding spherical joint center attached to the moving platform, and  $\mathbf{b}_i^0$  for the location of this latter center at the reference orientation of the moving platform.

The velocities of the prismatic joints, as a function of the angular velocities,  $\omega_x$  and  $\omega_y$ , can be expressed as

$$\begin{pmatrix} \dot{l}_1/l_1 \\ \dot{l}_2/l_2 \end{pmatrix} = \mathbf{F} \begin{pmatrix} \omega_x \\ \omega_y \\ 0 \end{pmatrix}, \quad (7.6)$$

where

$$\mathbf{F} = \begin{pmatrix} (\mathbf{a}_1 \times \mathbf{R}\mathbf{b}_1^0)^T \\ (\mathbf{a}_2 \times \mathbf{R}\mathbf{b}_2^0)^T \end{pmatrix}. \quad (7.7)$$

Then, singularities arise when  $\mathbf{F}$  is rank-deficient (see Chap. 6).

## 7.2 Three-Move Motion Planner

Given the initial and final poses represented by the rotation matrices  $\mathbf{R}_I$  and  $\mathbf{R}_F$ , respectively, the goal is to find three rotations,  $\mathbf{R}_x(\omega_1)$ ,  $\mathbf{R}_y(\omega_2)$ , and  $\mathbf{R}_x(\omega_3)$ , such that

$$\mathbf{R}_F = \mathbf{R}_x(\omega_3)\mathbf{R}_y(\omega_2)\mathbf{R}_x(\omega_1)\mathbf{R}_I, \quad (7.8)$$

That is,

$$\mathbf{R}_x(\omega_3)\mathbf{R}_y(\omega_2)\mathbf{R}_x(\omega_1) = \mathbf{R}_F\mathbf{R}_I^T = \mathbf{R}_T.$$

Therefore,  $\{\omega_1, \omega_2, \omega_3\}$  is a valid set of  $XYX$  Euler angles representing  $\mathbf{R}_T$ . As a consequence, if  $\omega_2 \in (0, \pi)$ , then

$$\omega_1 = \text{atan}2(r_{21}, -r_{31}), \quad (7.9)$$

$$\omega_2 = \arccos(r_{11}), \quad (7.10)$$

$$\omega_3 = \text{atan}2(r_{12}, r_{13}), \quad (7.11)$$

where  $r_{ij}$  stands for the  $(i, j)$  entry of  $\mathbf{R}_F$  (see, for example, [2] for the derivation of the above formulas).

The robot orientation as a function of time can be expressed as

$$\mathbf{R}(t) = \begin{cases} \mathbf{R}_x\left(\frac{t}{\Delta t}\omega_1\right)\mathbf{R}_I, & 0 \leq t < \Delta t \\ \mathbf{R}_y\left(\frac{t-\Delta t}{\Delta t}\omega_2\right)\mathbf{R}_x(\omega_1)\mathbf{R}_I, & \Delta t \leq t < 2\Delta t \\ \mathbf{R}_x\left(\frac{t-2\Delta t}{\Delta t}\omega_3\right)\mathbf{R}_y(\omega_2)\mathbf{R}_x(\omega_1)\mathbf{R}_I, & 2\Delta t \leq t \leq 3\Delta t \end{cases} \quad (7.12)$$

When particularized to the actuators arrangement studied in Chap. 6, this result leads to the motion planner presented there.

### 7.3 Two-Move Motion Planner

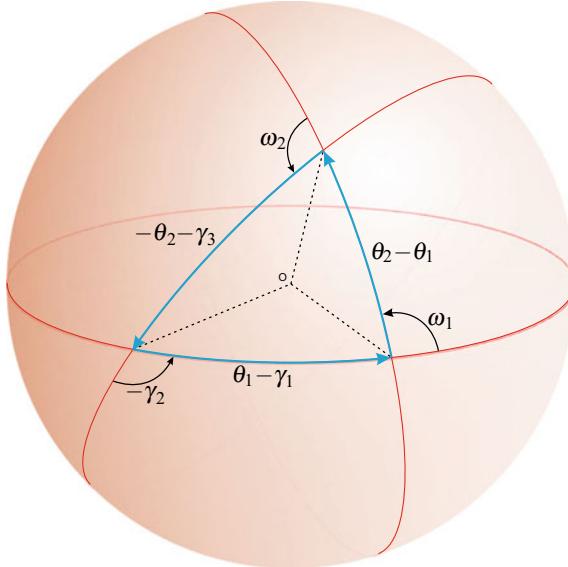
Given the initial and final poses,  $\mathbf{R}_I$  and  $\mathbf{R}_F$ , the goal is to find the angles  $\theta_1, \theta_2, \omega_1$  and  $\omega_2$  such that

$$\begin{aligned} \mathbf{R}_{n_1}(\omega_1)\mathbf{R}_{n_2}(\omega_2) &= \mathbf{R}_z(\theta_1)\mathbf{R}_x(\omega_1)\mathbf{R}_z(-\theta_1) \\ &\quad \mathbf{R}_z(\theta_2)\mathbf{R}_x(\omega_2)\mathbf{R}_z(-\theta_2) \\ &= \mathbf{R}_F\mathbf{R}_I^T = \mathbf{R}_T. \end{aligned} \quad (7.13)$$

It is always possible to express  $\mathbf{R}_T$  as follows

$$\mathbf{R}_T = \mathbf{R}_z(\gamma_1)\mathbf{R}_x(\gamma_2)\mathbf{R}_z(\gamma_3), \quad (7.14)$$

where  $\{\gamma_1, \gamma_2, \gamma_3\}$  is any set of valid  $ZXZ$  Euler angles representing  $\mathbf{R}_T$  (see [2] for explicit formulas). Then, substituting (7.14) in (7.13) and rearranging terms, we obtain



**Fig. 7.1** Spherical triangle whose closure equation is given by Eq.(7.15)

$$\begin{aligned} \mathbf{R}_z(\theta_1 - \gamma_1) \mathbf{R}_x(\omega_1) \mathbf{R}_z(\theta_2 - \theta_1) \mathbf{R}_x(\omega_2) \\ \mathbf{R}_z(-\theta_2 - \gamma_3) \mathbf{R}_x(-\gamma_2) = \mathbf{I}. \end{aligned} \quad (7.15)$$

This equation can be seen as the closure equation of the spherical triangle shown in Fig. 7.1 [9]. Then, the analogues of the law of cosines for the angles  $\omega_1$ ,  $\omega_2$  and  $-\gamma_2$  of this spherical triangle allow us to write

$$\begin{aligned} \cos(\theta_2 - \theta_1) \cos(\theta_1 - \gamma_1) - \cos(-\theta_2 - \gamma_3) \\ + \sin(\theta_2 - \theta_1) \sin(\theta_1 - \gamma_1) \cos(\pi - \omega_1) = 0, \end{aligned} \quad (7.16)$$

$$\begin{aligned} \cos(\theta_2 - \theta_1) \cos(-\theta_2 - \gamma_3) - \cos(\theta_1 - \gamma_1) \\ - \sin(\theta_2 - \theta_1) \sin(-\theta_2 - \gamma_3) \cos(\pi - \omega_2) = 0, \end{aligned} \quad (7.17)$$

$$\begin{aligned} \cos(\theta_1 - \gamma_1) \cos(-\theta_2 - \gamma_3) - \cos(\theta_2 - \theta_1) \\ + \sin(\theta_1 - \gamma_1) \sin(-\theta_2 - \gamma_3) \cos(\pi + \gamma_2) = 0, \end{aligned} \quad (7.18)$$

Equations (7.16) and (7.17) allow us to express  $\omega_1$  and  $\omega_2$  as a function of  $\theta_1$  and  $\theta_2$ , respectively, as follows

$$\omega_1 = \arccos \left( \frac{\cos(\theta_2 - \theta_1) \cos(\theta_1 - \gamma_1) - \cos(\theta_2 + \gamma_3)}{\sin(\theta_2 - \theta_1) \sin(\theta_1 - \gamma_1)} \right), \quad (7.19)$$

$$\omega_2 = \arccos \left( \frac{\cos(\theta_2 - \theta_1) \cos(\theta_2 + \gamma_3) - \cos(\theta_1 - \gamma_1)}{\sin(\theta_1 - \theta_2) \sin(\theta_2 + \gamma_3)} \right), \quad (7.20)$$

and Eq. (7.18), to express  $\theta_1$  as a function of  $\theta_2$  as

$$\theta_1 = \arctan \left( \frac{\cos \gamma_1 \cos \gamma_2 \sin(\theta_2 + \gamma_3) + \sin \gamma_1 \cos(\theta_2 + \gamma_3) - \sin \theta_2}{\sin \gamma_1 \cos \gamma_2 \sin(\theta_2 + \gamma_3) - \cos \gamma_1 \cos(\theta_2 + \gamma_3) + \cos \theta_2} \right). \quad (7.21)$$

Then, the robot orientation as a function of time is given by

$$\mathbf{R}(t) = \begin{cases} \mathbf{R}_{\mathbf{n}_1} \left( \frac{t}{\Delta t} \omega_1 \right) \mathbf{R}_I, & 0 \leq t < \Delta t \\ \mathbf{R}_{\mathbf{n}_2} \left( \frac{t-\Delta t}{\Delta t} \omega_2 \right) \mathbf{R}_{\mathbf{n}_1}(\omega_1) \mathbf{R}_I, & \Delta t \leq t \leq 2\Delta t \end{cases} \quad (7.22)$$

The motion generated by this two-move maneuver depends on a parameter:  $\theta_2$ . This permits to optimize the maneuver using different criteria. An obvious option is to choose the value of  $\theta_2$  that minimizes  $\omega_1^2 + \omega_2^2$ , i.e., the *total motion* of the sphere. This is exemplified in Sect. 7.5.

## 7.4 Single-Move Motion Planner

It was proved in [10] that, if  $\mathbf{W}(t)$  in (7.2) satisfies the differential equation

$$\dot{\mathbf{W}}(t) = \mathbf{N}_1 \mathbf{W}(t) - \mathbf{W}(t) \mathbf{N}_1, \quad (7.23)$$

then the solution to (7.2) can be expressed as

$$\mathbf{R}(t) = \exp(\mathbf{N}_1 t) \exp(\mathbf{N}_2 t) \mathbf{R}(0), \quad (7.24)$$

where  $\mathbf{N}_2 = \mathbf{W}(0) - \mathbf{N}_1$ .

The substitution of the ansatz

$$\mathbf{N}_1 = \begin{pmatrix} 0 & -\omega & 0 \\ \omega & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad (7.25)$$

in (7.23) yields the following system of equations

$$\begin{aligned} \dot{w}_y &= \omega w_x, \\ \dot{w}_x &= -\omega w_y, \end{aligned} \quad (7.26)$$

whose integration yields

$$\begin{aligned} w_x &= A \cos(\omega t + \omega_0), \\ w_y &= A \sin(\omega t + \omega_0). \end{aligned} \quad (7.27)$$

Then,

$$\mathbf{W}(0) = \begin{pmatrix} 0 & 0 & A \sin \omega_0 \\ 0 & 0 & -A \cos \omega_0 \\ -A \sin \omega_0 & A \cos \omega_0 & 0 \end{pmatrix}, \quad (7.28)$$

and, as a consequence,

$$\mathbf{N}_2 = \begin{pmatrix} 0 & \omega & A \sin \omega_0 \\ -\omega & 0 & -A \cos \omega_0 \\ -A \sin \omega_0 & A \cos \omega_0 & 0 \end{pmatrix}. \quad (7.29)$$

In order to use the above result to solve the problem of steering the moving platform from  $\mathbf{R}_I$  to  $\mathbf{R}_F$ , we can scale the time variable,  $t$ , so that the motion is completed at  $t = 1$ . Then, at  $t = 1$ , we have that

$$\mathbf{R}_F = \exp(\mathbf{N}_1) \exp(\mathbf{N}_2) \mathbf{R}_I = \mathbf{R}_z(\omega) \mathbf{R}_p(\delta) \mathbf{R}_I, \quad (7.30)$$

where

$$\mathbf{p} = \frac{1}{\sqrt{\omega^2 + A^2}} (A \cos \omega_0, A \sin \omega_0, -\omega)^T, \quad (7.31)$$

$$\delta = \sqrt{\omega^2 + A^2}. \quad (7.32)$$

In other words, the goal is to find  $\omega$ ,  $\omega_0$ , and  $A$  such that

$$\mathbf{R}_z(\omega) \mathbf{R}_p(\delta) = \mathbf{R}_F \mathbf{R}_I^T = \mathbf{R}_T. \quad (7.33)$$

If the vector of Euler parameters for the rotation defined by  $\mathbf{R}_T$  is  $(a, b, c, d)$ , it can be checked that the Euler parameters for  $\mathbf{R}_z(-\omega) \mathbf{R}_T$  can be expressed, as a function of  $\omega$ , as

$$a'(\omega) = a \cos \frac{\omega}{2} + d \sin \frac{\omega}{2}, \quad (7.34)$$

$$b'(\omega) = b \cos \frac{\omega}{2} + c \sin \frac{\omega}{2}, \quad (7.35)$$

$$c'(\omega) = c \cos \frac{\omega}{2} - b \sin \frac{\omega}{2}, \quad (7.36)$$

$$d'(\omega) = d \cos \frac{\omega}{2} - a \sin \frac{\omega}{2}. \quad (7.37)$$

Since, according to (7.33),  $\mathbf{R}_p(\delta) = \mathbf{R}_z(-\omega) \mathbf{R}_T$ , we have that

$$\cos \frac{\delta}{2} = a'(\omega), \quad (7.38)$$

$$\frac{A}{\delta} \cos \omega_0 \sin \frac{\delta}{2} = b'(\omega), \quad (7.39)$$

$$\frac{A}{\delta} \sin \omega_0 \sin \frac{\delta}{2} = c'(\omega), \quad (7.40)$$

$$-\frac{\omega}{\delta} \sin \frac{\delta}{2} = d'(\omega). \quad (7.41)$$

Now, observe that (7.38) and (7.41) depend only on  $\omega$  and  $\delta$ . From (7.38), we have that

$$\delta = \pm 2 \arccos(a'(\omega)). \quad (7.42)$$

Moreover, Eq. (7.38) can be rewritten as  $\sin(\delta/2) = \pm \sqrt{1 - [a'(\omega)]^2}$ . Then, dividing this expression by (7.41), we conclude that  $-\delta/\omega = \pm \sqrt{1 - [a'(\omega)]^2}/d'(\omega)$ . In other words,

$$\delta = \mp \omega \frac{\sqrt{1 - [a'(\omega)]^2}}{d'(\omega)}. \quad (7.43)$$

Therefore, equating (7.42) and (7.43) yields the following transcendental equation in  $\omega$

$$2d'(\omega) \arccos(a'(\omega)) + \omega \sqrt{1 - [a'(\omega)]^2} = 0. \quad (7.44)$$

Unfortunately, as it is usually the case for transcendental equations, no explicit solution has been found for (7.44). Thus, we have to rely at this point on a numerical method.

If we plot  $\delta$  as a function of  $\omega$  using (7.42) and (7.43), the intersection of both curves will correspond to the sought solutions. Figure 7.2 depicts a typical example of the obtained plots.

Observe that if  $(\delta, \omega)$  is a solution of (7.42) and (7.43), then  $(-\delta, \omega)$  is a solution as well, but they both correspond to the same physical motion. This simply accounts for the double covering of  $SO(3)$  when using Euler parameters.

Finally, with the obtained solutions for  $\omega$  and  $\delta$ , it is concluded from (7.32) that

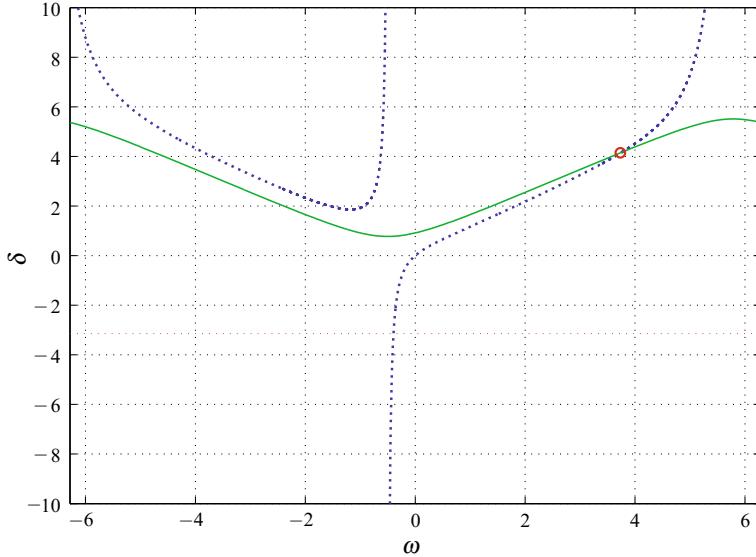
$$A = \sqrt{\delta^2 - \omega^2}, \quad (7.45)$$

and, from (7.40) and (7.39), that

$$\omega_0 = \arctan \left( \frac{c'(\omega)}{b'(\omega)} \right). \quad (7.46)$$

In conclusion, the robot orientation as a function of time can simply be expressed as

$$\mathbf{R}(t) = \mathbf{R}_z \left( \frac{t}{\Delta t} \omega \right) \mathbf{R}_p \left( \frac{t}{\Delta t} \delta \right) \mathbf{R}_I, \quad 0 \leq t \leq \Delta t. \quad (7.47)$$



**Fig. 7.2** Typical example of plots representing the positive branches of (7.43) and (7.42) shown in dashed blue and solid green lines, respectively

It can be verified that the equivalent axis of rotation of  $\mathbf{R}_z\left(\frac{t}{\Delta t}\omega\right)\mathbf{R}_p\left(\frac{t}{\Delta t}\delta\right)$  lies on the XY plane for any value of  $t$ , as required [4, p.97].

## 7.5 Example

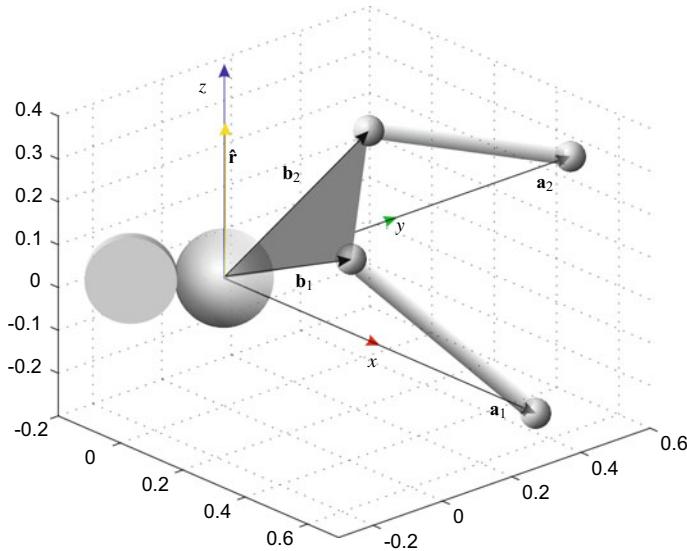
Let us consider the non-holonomic orienting robot shown in Fig. 7.3. The center of the sphere is located at the origin, the spherical joints attached to the base are centered at  $\mathbf{a}_1 = (1, 0, 0)^T$  and  $\mathbf{a}_2 = (0, 1, 0)^T$ , and those attached to the moving platform, in the reference orientation, at  $\mathbf{b}_1^0 = (0.5, 0, 0)^T$  and  $\mathbf{b}_2^0 = (0, 0.5, 0)^T$ . Due to the non-holonomic constraint, the sphere cannot rotate about  $\hat{\mathbf{r}}$ , which is assumed to be aligned with the  $z$ -axis, as above.

Let us also assume that the initial and final orientations of the moving platform are given by

$$\mathbf{R}_I = \begin{pmatrix} 0.7513 & -0.5275 & -0.3965 \\ 0.5071 & 0.8460 & -0.1646 \\ 0.4223 & -0.0774 & 0.9031 \end{pmatrix}, \quad (7.48)$$

and

$$\mathbf{R}_F = \begin{pmatrix} 0.9970 & 0.0534 & 0.0553 \\ -0.0192 & 0.8700 & -0.4927 \\ -0.0744 & 0.4902 & 0.8684 \end{pmatrix}, \quad (7.49)$$



**Fig. 7.3** Location of the joints and the disk for the robot used in the example

respectively. Then,

$$\mathbf{R}_T = \mathbf{R}_F \mathbf{R}_I^{-1} = \begin{pmatrix} 0.6991 & 0.5416 & 0.4668 \\ -0.2780 & 0.8074 & -0.5204 \\ -0.6588 & 0.2340 & 0.7150 \end{pmatrix}. \quad (7.50)$$

### 7.5.1 Three-Move Motion Planner

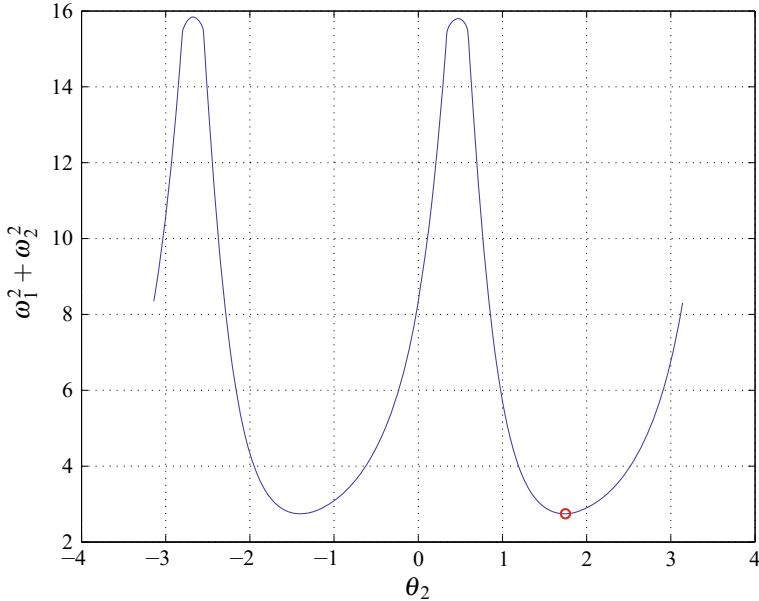
The three-step maneuver results from applying Eqs. (7.9), (7.10) and (7.11). This yields

$$\omega_1 = 0.8594, \quad \omega_2 = 0.7967, \quad \omega_3 = -0.3993.$$

A representation of the robot motion following the resulting trajectory can be seen in Fig. 7.5 (top row).

### 7.5.2 Two-Move Motion Planner

To obtain a two-move maneuver, the first step is to compute a set of values for  $\gamma_1$ ,  $\gamma_2$ , and  $\gamma_3$  satisfying (7.14). Then, we can express  $\omega_1$  and  $\omega_2$  as a function of  $\theta_2$  using (7.19), (7.20), and (7.21). Since we can arbitrarily choose  $\theta_2$ , we can pick out



**Fig. 7.4** The minimum of  $\omega_1^2 + \omega_2^2$  as a function of  $\theta_2$  is attained at 1.7384. This value determines the two-move maneuver used in the example

the value that minimizes  $\omega_1^2 + \omega_2^2$ . This minimum is attained at  $\theta_2 = 1.7384$  (see Fig. 7.4). The substitution of this value in (7.21) yields  $\theta_1 = 2.5829$ . Then, the axes of rotation are

$$\mathbf{n}_1 = (-0.848, 0.530, 0)^T, \quad \mathbf{n}_2 = (-0.167, 0.986, 0)^T,$$

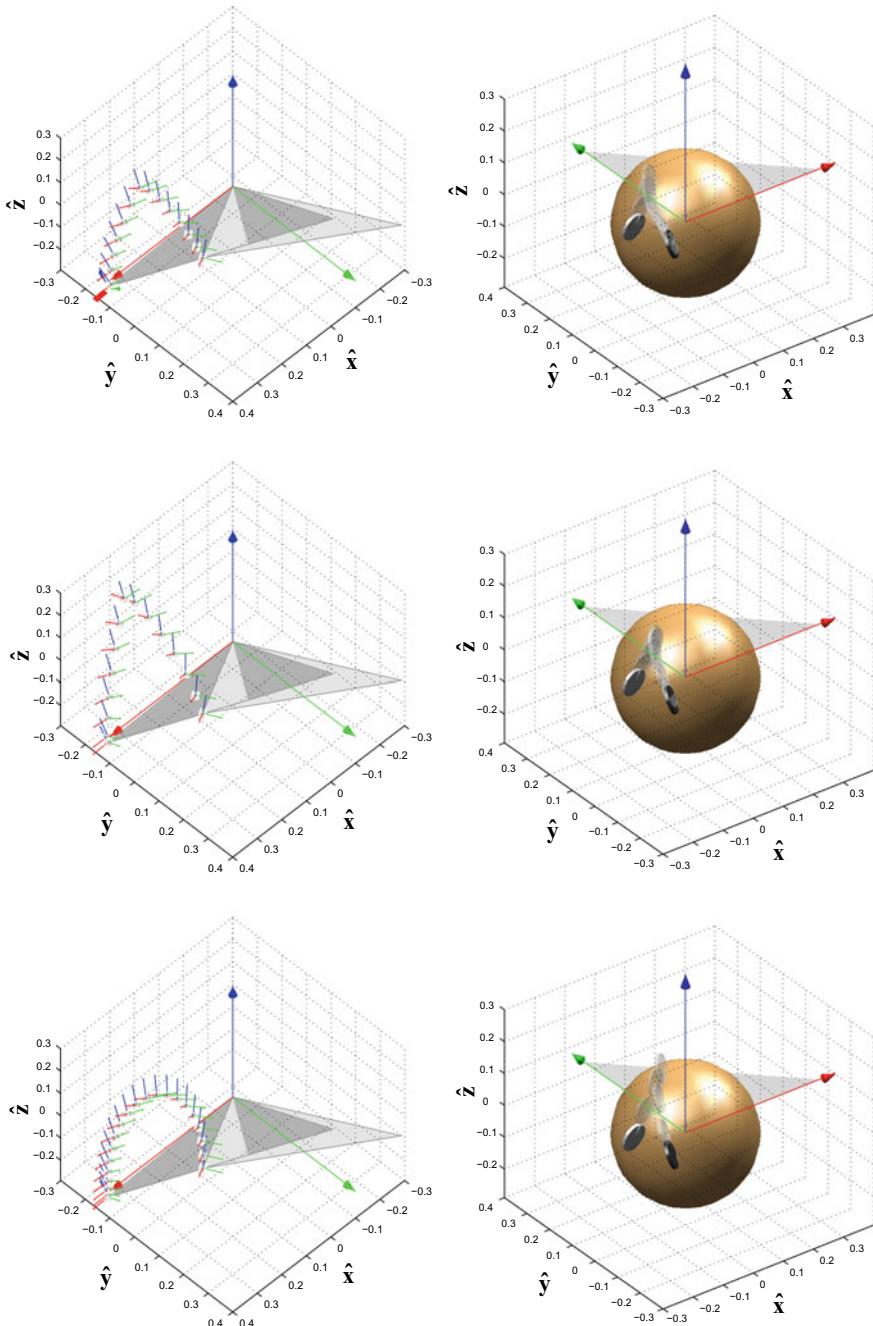
and the rotated angles about them are obtained substituting the values of  $\theta_1$  and  $\theta_2$  in (7.19) and (7.20), respectively. This yields

$$\omega_1 = 1.1713, \quad \omega_2 = -1.1708.$$

A representation of the robot motion following the resulting trajectory can be seen in Fig. 7.5 (center row).

### 7.5.3 Single-Move Motion Planner

Following the procedure detailed in Sect. 7.4, the first step consists in computing the Euler parameters for the rotation defined by (7.50). The result is



**Fig. 7.5** Example of the motions generated by the three motion planners shown with respect to the reference frames of the base and of the moving platform (left and right column, respectively)

$$\begin{aligned}a &= 0.8974, \\b &= 0.2102, \\c &= 0.3136, \\d &= -0.2283.\end{aligned}$$

Then, we can plot  $\delta$  as a function of  $\omega$  using both (7.42) and (7.43). The result is plotted in Fig. 7.2. The intersection of both curves occurs at

$$\delta = 4.1578, \quad \omega = 3.7496.$$

Then, the substitution these values in (7.45) and (7.46) yields

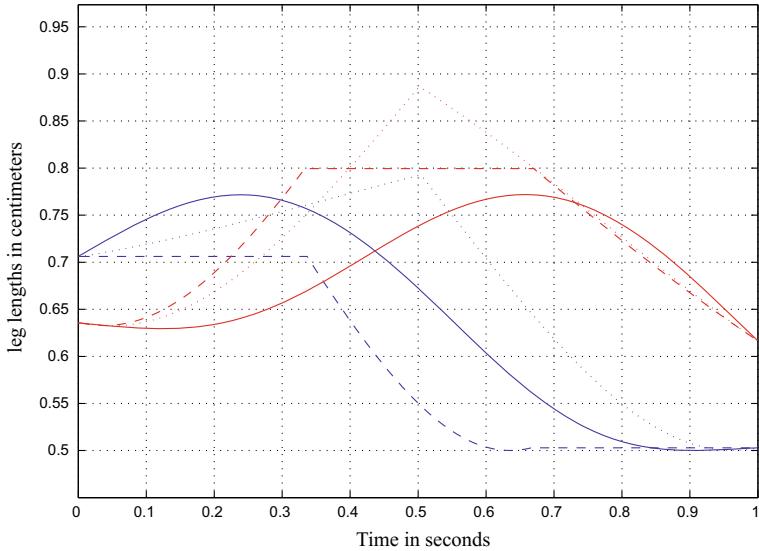
$$A = 1.7965, \quad \omega_0 = -0.8945.$$

A representation of the motion followed by the robot along the resulting path can be seen in Fig. 7.5 (bottom row).

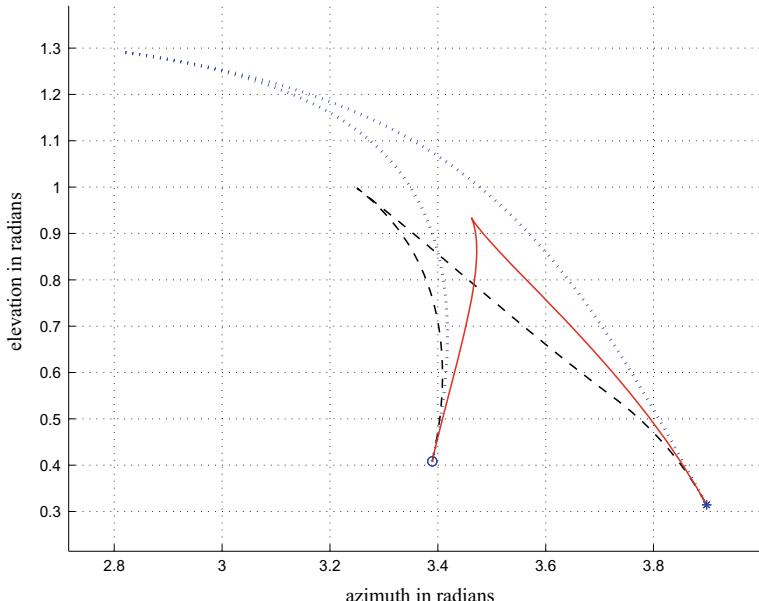
#### 7.5.4 Comparing the Three Motion Planners

Figure 7.5 shows the motion generated by the three motion planners organized in three rows. From top to bottom we can see the motions ordered in decreasing number of moves. In the left column, we have light gray and dark gray triangles representing the moving platform in its initial and final orientation, respectively. The sequence of reference frames illustrate the path followed by  $\mathbf{b}_1$ . This is the motion generated as seen from the base reference frame. If we fix the observer to the moving reference frame, the motion followed by the disk on the sphere is better appreciated. This is represented in the right column.

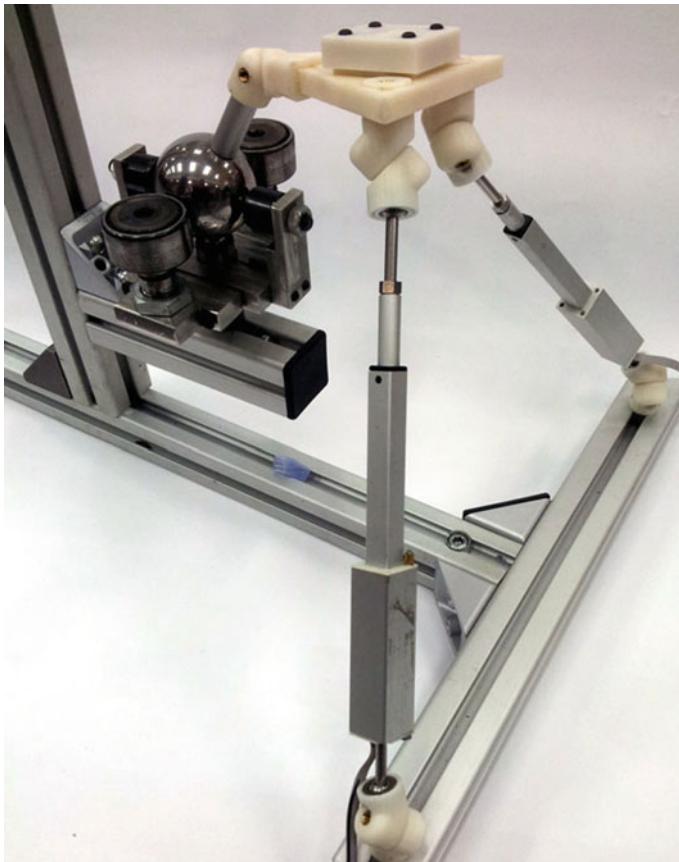
All three trajectories behave well and quite similarly in this example. A greater difference is observed when translating the generated motions into a variation of the two leg lengths using (7.5). The result is represented in Fig. 7.6. The single-move motion planner generates a differentiable path in the space of actuation variables. It can lead to faster motions because the generated trajectories do not contain zero-velocity points, thus making the use of the studied parallel robot possible in a larger range of applications. Finally, it is worth noting that, although using the single-move motion planner the time variation of the legs' lengths are differentiable, the path followed by the point of contact between the disk and the sphere contains cusp points (see Fig. 7.7). These points play a fundamental role in non-holonomic motion planners [7]. An important result is that they are here automatically generated.



**Fig. 7.6** Temporal variation of the leg lengths ( $l_1$  in blue and  $l_2$  in red) for the three generated trajectories. The leg lengths for the three-move maneuver are represented in dashed lines, for the two-move maneuver, in dotted lines, and for the single-move maneuver, in solid lines



**Fig. 7.7** Paths followed by the point of contact between the disk and the sphere for the three motion planners. The initial and final configurations are indicated by ‘o’ and ‘\*’, respectively. The same line code as in Fig. 7.6 is used to indicate the results generated by the different motion planners



**Fig. 7.8** Experimental testbed

## 7.6 Implementation

To validate the presented motion planners, the testbed shown in Fig. 7.8 was implemented. It features a redimensionable structure, two SPS legs and a moving platform fixed to a non-holonomic joint. The structure is constructed using aluminum profiles with the idea of being modular such that the S joints and the non-holonomic joint can be placed where desired. Modularity also includes the ability to re-orient the non-holonomic constraint to point in any direction.

Initially, each SPS leg had as prismatic actuators a miniature Firgelli linear actuators with a stroke of 100 mm [3]. Due to their slowness and inaccuracies, they were replaced by Dynamixel MX64 motors with some extra hardware. The legs are attached to the base through spherical joints based on spherical bearings housed in 3D printed plastic receptacles which enables the relocation of the joints.

The non-holonomic joint is based on a steel ball. It has been perforated using an electric discharge machine. Its motion is constrained by two rollers arranged in opposing positions from the center. Three free-rolling spheres are used to keep the joint centered in the plane perpendicular to the line between the contact points of the rollers (see Sect. 3.2 for a detailed explanation of the design and construction of this non-holonomic joint).

The orientation of the platform is measured using a Phidget spatial 3/3/3 sensor that communicates via USB [8]. This board has a 3-axis accelerometer, gyroscope and compass. Motors and sensor are interfaced through USB using a software developed for interfacing with Matlab. The software enables reading the platform orientation and the leg lengths, and commanding the motors.

The motion planning problem of an S<sub>n</sub>-2UPS parallel orienting robot has been solved first using a three-move motion planner. Then, we have observed that it can be simplified by using only two moves. Finally, we have arrived at the rather surprising result that a single-move path planner generates the correct motor commands, in general, to reach an arbitrary final configuration within the robot's workspace. That is, the analyzed orienting robot cannot follow arbitrary paths connecting two orientations, but it is possible to find feasible smooth paths in the actuators variables which allow connecting two arbitrary orientations for the moving platform. We do not know of any other non-trivial non-holonomic motion planning problem for which such kind of solution have been found. Finally, it is also important to mention that the obtained smooth paths result from choosing the ansatz in (7.25), but there are other alternatives. This means that there are probably infinitely many  $L^\infty$  paths in the space of actuation variables that satisfy the non-holonomic constraint. This opens the possibility of optimizing the path according to some criterion, or even the possibility of finding paths that avoid the need of a numerical method.

## References

1. R. Di Gregorio, Position analysis, path planning, and kinetostatics of single-loop RU-(nS)PU wrists. *Mech. Mach. Theory* **74**, 117–133 (2014)
2. D. Eberly, Euler Angle Formulas, Geometric Tools, LLC (2007), <http://www.geometrictools.com/>
3. Firgelli Technologies Inc. Innovator in the Micro-motion (2015), <http://www.firgelli.com/>
4. P. Grosch, Parallel Robots with unconventional joints to achieve under-actuation and reconfigurability. Ph.D. thesis, Polytechnic University of Catalonia, Spain (2016)
5. P. Grosch, F. Thomas, Geometric path planning without maneuvers for non-holonomic parallel orienting Robots. *IEEE Robot. Autom. Lett.* **1**(2), 1066–1072 (2016)
6. J. Jakubiak, W. Magiera, K. Tchoń, Control and motion planning of a non-holonomic parallel orienting platform. *J. Mech. Robot.* **7**(4) (2015)
7. J.P. Laumond, S. Sekhavat, F. Lamiraux, Guidelines in nonholonomic motion planning for mobile Robots. *Robot Motion Plan. Control* **229**, 1–53 (1998)
8. Phidgets Co. Products for USB Sensing and Control (2015), <http://www.phidgets.com/>

9. C.W. Wampler, Displacement analysis of spherical mechanisms having three or fewer loops. ASME J. Mech. Des. **126**(1), 93–100 (2004)
10. M.Y. Wu, Some new results in linear time-varying systems. IEEE Trans. Autom. Control **20**(1), 159–161 (1975)

# Chapter 8

## Conclusions



It has been shown how the substitution of ordinary joints with lockable revolute joints ( $R_b$  joints) or non-holonomic spherical joints ( $S_n$  joints), which we have referred here to as unconventional joints, provide lower-mobility parallel robots with interesting features. One important common characteristic of this kind of robots is that they can approximate, in general, trajectories for their moving platform in a configuration space of dimension higher than the number of their continuously actuated joints. In this monograph, the kinematics of the following three robots have been analyzed: the  $4R_bRPS$  spatial reconfigurable robot (Chap. 2), the  $3S_nPU$  spatial non-holonomic robot (Chaps. 4 and 5), and the  $S-2SPS$  spherical non-holonomic robot (Chaps. 6 and 7).

The analysis of each of these new robots has been conducted with the greatest possible generality by including in most cases the computation of its direct and inverse kinematics (both in position and velocity), the analysis of its singularities, the computation of its workspace, the design and implementation of a motion planning algorithm, and the implementation of a prototype to validate the theoretical results.

All the new designs have been derived from the celebrated Gough-Stewart platform (including its spherical counterpart), to which, through certain geometric transformations, some of its joints are replaced by lockable joints or non-holonomic joints. These substitutions permit reducing the number of legs (and hence of actuators) without losing the robot's ability to bring its mobile platform to any position and orientation within its workspace. In general, these new designs have a larger workspace compared to the Gough-Stewart platform from which they are derived, as the possibility of collisions between legs is reduced, and a lower weight and cost due to the reduction in the number of legs and actuators. These advantages do not come without a cost: it is necessary, in all cases, to plan maneuvers to reach the desired position and orientation for the moving platform. Therefore, the obtained robots will only be suitable for applications where no high speeds are needed, accuracy is

required in the final pose, and a certain margin of error is acceptable in the generated trajectories. These applications include most positioning and point-to-point tasks.

In the studied cases, it has been possible to design an algorithm to automatically generate the necessary maneuvers to connect two arbitrary configurations. The nature of the presented algorithms ranges from pure geometrical to entirely analytical. Indeed, while the motion planning problem for parallel platforms with lockable joints is a geometrical and combinatorial problem, the same problem for platforms with non-holonomic joints requires, in general, the use of differential geometry tools. The presented methods solve non-trivial cases, thus being of great value as references for other parallel robots with the considered unconventional joints.

The new designs can also be seen as new “benchmarks” in the area of non-holonomic and reconfigurable systems where to test new algorithms for solving motion planning problems. In particular, the proposed new non-holonomic systems enlarge the set of commonly used non-holonomic systems used as examples, which was limited to systems like vehicles, possibly with arms, or vehicles dragging n-carts in series, some of them having more academic than practical interest.

The  $4R_bRPS$  parallel platform has been proved to be able to move its moving platform in a six-dimensional operational space by using only four actuators. Maneuvers essentially consist, in this case, in changing the set of locked joints at via configurations. It has been shown that the implemented motion planner is also useful to avoid collisions and singularities. We conjecture that, by properly locating the joints, it would be possible to avoid all singularities. If this conjecture is eventually proved, this parallel platform would attain a remarkable place in the universe of parallel robots. It is worth noting with reference to this robot that, following the same reasoning that has led us to derive it from the Gough-Stewart platform, it would be possible to replace the  $UPS$  legs of a spherical  $S-3UPS$  parallel robot with  $R_bRPS$  legs to derive a reconfigurable  $S-2R_bRPS$  spherical robot whose moving platform can be moved in a three-dimensional configuration space by using only two actuators and by introducing via poses, configurations where the locked  $R_b$  pair is changed. The analysis of such a robot deserves some attention.

The  $3S_nPU$  parallel robot is a complex non-holonomic system that, thanks to a compact formulation of its instantaneous kinematics, an elegant motion planning algorithm has been derived. This algorithm can be applied only off-line. Speeding up the algorithm may be achieved by a suitable program code optimization and by using a dedicated hardware, but likely it will also require a simplification of the mathematical model. This opens up another avenue of future research.

The  $S_n-2UPS$  parallel orienting platform has been proved to be able to move its moving platform in the three-dimensional space of orientations by using only two actuators. Based on a rather old result on linear time-varying systems, it has been shown that a single-step path planner allows this robot, in general, to move from one configuration to any other in its workspace by following an infinitely differentiable path in the actuator space variables.

The presented motion planners for the  $S_n-2UPS$  robot are open-loop methods. However, it would be desirable to construct the input as a function of the system state to compensate for noises and errors in the system. Motion planners that generate

maneuvers with zero-velocity points cannot be translated into control systems in an obvious way. The situation changes with the presented single-move path planner. This is certainly a point that deserves further attention. The result would be a practical algorithm for planning and controlling the motions of the studied platform that can help to achieve all its potential benefits. The presented ideas seem to be applicable to other non-holonomic mechanical systems whose orientation has to be controlled.